



Towards a realistic analysis of sorting and searching algorithms

Thu Hien Nguyen Thi

► To cite this version:

Thu Hien Nguyen Thi. Towards a realistic analysis of sorting and searching algorithms. Computer Science [cs]. Université de Caen Basse-Normandie, 2014. English. NNT: . tel-01134104

HAL Id: tel-01134104

<https://hal.science/tel-01134104>

Submitted on 22 Mar 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ÉCOLE DOCTORALE SIMEM

T H È S E

Towards a realistic analysis of sorting and searching algorithms

Présentée par Thu-Hien NGUYEN-THI
et soutenue le 9 décembre 2014
en vue de l'obtention du

Doctorat de l'Université de Caen

Spécialité : Informatique
(Arrêté du 30 mars 1992)

Composition du jury :

Rapporteurs: Conrado MARTINEZ, Professeur à l'Universitat Politècnica de Catalunya
Cyril NICAUD, Professeur à l'Université Paris-Est, Marne-la-Vallée

Examineurs: Julien CLÉMENT, Chargé de recherche au CNRS, Université de Caen, (Co-directeur)
Philippe DUCHON, Professeur à l'Université Bordeaux 1
Danièle GARDY, Professeure à l'Université de Versailles, Saint-Quentin
Hsien-Kuei HWANG, Directeur de recherche à Academia Sinica (Taiwan)
Brigitte VALLÉE, Directrice de recherche au CNRS, Université de Caen (Directrice)

ÉCOLE DOCTORALE SIMEM

T H È S E

Towards a realistic analysis of sorting and searching algorithms

Présentée par Thu-Hien NGUYEN-THI
et soutenue le 9 décembre 2014
en vue de l'obtention du

Doctorat de l'Université de Caen

Spécialité : Informatique
(Arrêté du 30 mars 1992)

Composition du jury :

Rapporteurs: Conrado MARTINEZ, Professeur à l'Universitat Politècnica de Catalunya
Cyril NICAUD, Professeur à l'Université Paris-Est, Marne-la-Vallée

Examineurs: Julien CLÉMENT, Chargé de recherche au CNRS, Université de Caen, (Co-directeur)
Philippe DUCHON, Professeur à l'Université Bordeaux 1
Danièle GARDY, Professeure à l'Université de Versailles, Saint-Quentin
Hsien-Kuei HWANG, Directeur de recherche à Academia Sinica (Taiwan)
Brigitte VALLÉE, Directrice de recherche au CNRS, Université de Caen (Directrice)

Remerciements

Je souhaite remercier en premier lieu ma directrice et mon directeur de thèse qui m'ont beaucoup soutenu pendant ce travail doctoral. Sans eux, je n'aurais pu mener cette thèse à son terme.

Je tiens à exprimer mes plus vifs remerciements à Brigitte Vallée, ma directrice de thèse pour la confiance qu'elle m'a accordée en me proposant ce sujet de thèse à l'issue de mon Master 2. Je la remercie pour ses multiples conseils et tous les heures qu'elle a consacrées à diriger cette recherche. Sa compétence, sa rigueur scientifique et sa clairvoyance m'ont beaucoup appris. Ils ont été et resteront avec moi mes moteurs de travail.

Je souhaite exprimer ma gratitude à Julien Clément, mon directeur de thèse, qui n'a jamais cessé de m'encourager et de m'aider pendant la thèse. Je suis extrêmement sensible par ses qualités d'humaine d'écoute et de compréhension tout au long de la thèse. Il m'a transmis tant de plaisir et d'envie de faire la recherche que, sans lui, cette thèse n'aurait pas été aussi intéressante et productive.

Conrado Martínez et Cyril Nicaud m'ont fait l'honneur d'accepter d'être mes rapporteurs. Je les remercie vivement pour le temps qu'ils m'ont accordé et pour les remarques et suggestions qu'ils m'ont proposées.

Je remercie sincèrement Philippe Duchon, Danièle Gardy, Hsien-Kuei Hwang pour avoir acceptés d'être membres de mon jury.

Je tiens à remercier tous les membres du GREYC et en particulier, les membres de l'équipe Amacc qui m'ont accueilli dans le laboratoire. Plus personnellement, je remercie Loïck, Ali, Arielle, Virginie, Véronique pour leur aide. Je n'oublie pas de remercier l'équipe d'administrateurs de réseaux, en particulier, Véronique, Pierre, Davy pour leur compagnie. Mes remerciements vont également à tous mes amis à Caen et dans le laboratoire donc Mathieu, Laura, Morgan, Kanal...

Finalement, j'adresse mes sentiments intimes mêlés d'une grande gratitude à ma famille dont mes parents, mon frère et mes soeurs pour leur support volontaire dans ma vie. Merci de tout mon coeur à mon mari et à notre petit Pooh qui vient d'avoir cinq mois et qui est toujours notre source d'amour et bonheur.

Contents

Remerciements	i
Introduction	1
1 Sorting and searching algorithms.	9
Analyses in the key model.	9
1.1 Algorithms and analysis of algorithms.	10
1.1.1 A general introduction	10
1.1.2 Sorting and searching algorithms: the input key model.	11
1.1.3 Sorting and searching algorithms: the costs of interest.	12
1.1.4 Important parameters of permutations.	13
1.1.5 A lower bound for sorting in the key model	14
1.1.6 The main algorithms under study.	16
1.2 The QuickSort algorithm.	16
1.2.1 Description	16
1.2.2 Mean number of key-comparisons.	17
1.2.3 Mean number of local comparaisons.	19
1.3 The QuickSelect algorithm and its variants	20
1.3.1 Description	20
1.3.2 Mean number of key comparisons.	20
1.3.3 Mean number of local key-comparisons.	24
1.4 The InsertionSort algorithm.	25
1.4.1 Description	25
1.4.2 Mean number of key comparaisons	26
1.4.3 Mean number of local key comparaisons.	27
1.5 The BubbleSort algorithm	27
1.5.1 Description.	27
1.5.2 Mean number of comparisons.	28
1.5.3 Mean number of local key comparisons.	29
1.6 The Selection-Minimum Algorithm	30
1.7 Towards a realistic analysis of selection and sorting algorithms.	31
1.7.1 Conclusion on the mean number of local comparaisons.	31
1.7.2 Normalized and non normalized algorithms.	31
2 Sources	35
2.1 General sources	36
2.1.1 Generating functions of the source.	36
2.1.2 Entropy.	37
2.1.3 Coincidence.	37

2.2	Simple sources.	38
2.2.1	Memoryless sources.	38
2.2.2	Markov chains.	39
2.2.3	Other instances of “simple” sources: intermittent sources.	41
2.3	Dynamical sources	42
2.3.1	Definition of a dynamical source	42
2.3.2	Simple sources seen as dynamical sources	44
2.3.3	Transfer operators	46
2.4	Parametrization of a general source.	48
2.4.1	Non ambiguous sources.	49
2.4.2	Parametrization of a general source.	49
2.4.3	Geometry of the source	50
3	Tameness of Sources	53
3.1	Properties on the line $\{\Re s = 1\}$. (Quasi-)periodicity.	54
3.1.1	Case of a memoryless source.	54
3.1.2	Case of a Markov chain.	57
3.1.3	Case of a dynamical source.	58
3.1.4	Quasi periodicity is exceptional.	59
3.2	Properties on the left half-plane $\{\Re s < 1\}$. Tameness.	60
3.2.1	Shape of tameness regions.	60
3.2.2	Three types of tameness	60
3.2.3	Tameness in vertical strips “with holes”. The periodic case.	61
3.2.4	Tameness in vertical strips. The UNI condition.	61
3.2.5	Tameness in hyperbolic regions. Diophantine conditions.	62
3.2.6	A small piece of history.	64
3.2.7	Conclusion of the study for classical sources.	64
3.3	Weak tameness of source.	64
4	Presentation of the main tools	67
4.1	Poisson and Bernoulli models	68
4.1.1	Definition of the models.	68
4.1.2	Poisson model.	68
4.1.3	Generating functions.	70
4.1.4	Return from the Poisson model to the Bernoulli model.	70
4.2	Depoissonization.	71
4.2.1	A general depoissonization result.	71
4.2.2	The \mathcal{JS} conditions.	71
4.2.3	Analytic extension of the sequence $T(n)$	72
4.3	Binomial recurrence and Rice method	72
4.3.1	Binomial recurrence.	72
4.3.2	First step: An integral form.	72
4.3.3	Step 2. Shifting to the left.	74
4.3.4	Tameness of $\varpi(s)$	74
4.3.5	Asymptotic estimates for $T(n)$	75
4.4	Proof of the asymptotic estimates in the Rice method.	76
4.4.1	The statement	76
4.4.2	Estimates near the real axis.	77
4.4.3	Far from the real axis.	77
4.4.4	Proof of Proposition 4.9.	78

4.5	Mellin transform	79
4.5.1	Definition of the Mellin transform	79
4.5.2	Main properties of the Mellin transform.	80
4.5.3	Inverse Mellin transform	81
4.5.4	Asymptotic expansion of f with its Mellin transform f^*	81
4.6	The “Poisson–Mellin–Newton–Rice” cycle.	82
4.6.1	The Valuation-Degree Condition	82
4.6.2	Existence and expression of the analytic lifting $\varpi(s)$	82
5	General framework for a more realistic analysis	85
5.1	Main principles.	86
5.1.1	Two probabilistic models for the inputs of the algorithm.	86
5.1.2	The cost of interest.	86
5.1.3	A general scheme for the analysis.	87
5.1.4	Four steps in the analysis.	88
5.1.5	Main objects in the analysis.	88
5.1.6	General Results. The case of the SelMin Algorithm.	89
5.2	Expression of the Poissonized density.	90
5.2.1	Mean number of comparisons between two keys.	90
5.2.2	Mean number of comparisons between two parametrized words.	91
5.2.3	Density of an algorithm – A first expression.	91
5.2.4	Expression of the density in the Poisson model via its coefficients. Automatic transfer.	92
5.3	Exact mean number of comparisons.	94
5.3.1	Density and symbol-density.	94
5.3.2	Coefficients of the mean number of symbol comparisons in the Poisson model	94
5.3.3	Exact expression of the mean number of symbol-comparisons in the Bernoulli model.	95
5.4	Analytic study of the mixed Dirichlet series.	96
5.4.1	Using Rice method.	96
5.4.2	Tameness of the mixed Dirichlet series.	96
5.4.3	Tameness transfer.	97
5.4.4	Tameness of the mixed Dirichlet series.	97
5.4.5	Singular expression of the mixed Dirichlet series at $s = \sigma_0$	98
5.4.6	Asymptotic expansions for $S(n)$	98
5.5	Asymptotic study in the SelMin case.	98
5.5.1	Main result.	98
5.5.2	Study on the halfplane $\Re s \geq 2$	99
5.5.3	Study on the vertical strip $\Re s \in [1, 2]$	100
5.5.4	Situation on the left of the vertical line $\Re s = 1$	100
6	Towards a realistic analysis of three sorting algorithms.	
	Lower bounds for the digital cost.	103
6.1	The two main results	104
6.1.1	Tameness properties of the source.	104
6.1.2	The main result.	105
6.1.3	Main constants of interest.	105
6.1.4	Beyond Λ -tameness?	106
6.1.5	Robustness.	107

6.1.6	An asymptotic lower bound for $S(n)$.	107
6.2	Mean number of symbol comparisons.	107
6.2.1	Expression for $\pi_n(i, j)$	107
6.2.2	Expression of $\hat{\pi}[u, t]$ in the basis G_m .	107
6.2.3	Expression of the coefficient $\varphi(k, u, t)$	108
6.2.4	Expressions of the mixed Dirichlet series $\varpi(s)$.	108
6.2.5	Tameness of the mixed Dirichlet series.	109
6.2.6	Mean number of symbol comparisons	111
6.3	An alternative approach.	111
6.3.1	Seidel's point of view.	112
6.3.2	Seidel's result	112
6.3.3	Faithfulness.	113
6.3.4	A relation between faithful sorting algorithms and trie parameters.	114
6.3.5	Analysis of additive trie parameters.	116
6.3.6	Proof of Proposition 6.11.	117
6.3.7	An alternative proof for QuickSort and InsSort .	119
6.4	An asymptotic lower bound for $S(n)$.	120
6.4.1	Proof of Theorem 6.5.	120
6.4.2	Proof of Lemma 6.13	121
6.4.3	Relation between various lower bounds.	122
7	Towards a realistic analysis of the QuickSelect algorithm.	123
7.1	Results	124
7.1.1	Tameness properties needed for the source.	124
7.1.2	General result.	125
7.1.3	Constants of the analysis.	125
7.1.4	Particular case of the binary source.	126
7.2	Algebraic analysis.	128
7.2.1	Generalities	128
7.2.2	When the keys are words.	128
7.2.3	The QuickVal algorithm	129
7.2.4	Poissonized density.	130
7.3	Expectation in the Poisson model.	131
7.3.1	Exact formulae for the mean number of symbol comparisons in the Bernoulli model.	131
7.3.2	The case of the algorithm QuickQuant .	132
7.3.3	General principles for the Asymptotic Analysis	133
7.4	Asymptotic Analysis of QuickVal	133
7.4.1	Analytic lifting and Rice formula.	133
7.4.2	Asymptotic estimates for the mean number of symbol comparisons for QuickMin , QuickMax , QuickVal and QuickRand .	134
7.4.3	Tameness of the mixed series $\varpi(s)$.	134
7.4.4	Principles of the proof of Proposition 7.12.	135
7.4.5	Study of integrals over triangles.	136
7.4.6	Study of the integral over the rectangle \mathcal{R} .	137
7.4.7	End of the proof.	139
7.5	Return to the analysis of the QuickQuant $_{\alpha}$ Algorithm.	139
7.5.1	Regularity of the asymptotic slope $\alpha \mapsto \rho(\alpha)$ of QuickVal .	139
7.5.2	Return to the analysis of QuickQuant .	141
7.5.3	Laplace's method.	142

7.5.4	A tailored version of the Laplace method.	143
7.5.5	Proof of Theorem 7.16.	144
7.6	Study of the asymptotic slope $\rho(\alpha)$	146
7.6.1	An alternative expression for the asymptotic slope $\alpha \mapsto \rho(\alpha)$ of QuickVal.	146
7.6.2	Maximality of the asymptotic slope of the QuickVal algorithm at $\alpha = 1/2$	147
8	Conclusion and perspectives	151
A	Generating functions	155
A.1	Generating function	155
A.1.1	Definitions	155
A.1.2	Symbolic approach	156
B	Complex analysis	159
B.1	Holomorphic function and Cauchy theorem	159
B.2	Meromorphic functions, poles and residue theorem	162
C	Upper and lower bounds	165
	Abstract	173

Introduction

History and motivations

This thesis begins with the works of Clément, Fill, Flajolet and Vallée [12, 13]. There are two main classes of sorting and searching algorithms: those which work with keys and those which work with words (or strings). For a lot of basic (sorting and searching) algorithms, a key is seen as a whole “atomic” structure and its representation is not taken into account. By opposition the structure of a word as a sequence of symbols is essential in algorithms on strings. The unit cost of algorithms in the first class is the comparison between keys while, in the second class, it is the comparison between symbols. Thus, it seems unfair in general to compare algorithms if we do not use the same unit cost. In 1998, Sedgewick proposed to analyze basic algorithms (sorting and searching algorithms) when dealing with words and not “atomic” keys. In this case, the realistic cost of comparing two words is the number of symbol comparisons needed to distinguish them in lexicographic order. This cost is closely related to the length of the longest common prefix of two words, called here the coincidence. We wish in this thesis to provide a fairer comparison between symbol-based algorithms (like `radix-sort` or algorithms using digital trees or tries) and key-based algorithms (`QuickSort`, `QuickSelect`, etc.)

General context of the thesis

This thesis belongs to three domains in theoretical computer science: Algorithmics, Information Theory and Analytic Combinatorics.

Algorithmics. As the title says, the objective of the thesis is the analysis of sorting and searching algorithms. The algorithms `QuickSort` and `QuickSelect`, invented by Hoare in 1962 [41], are important, well-known and representative of the class of *divide and conquer algorithms*. They appear in many algorithmic textbooks, for example, those of Knuth [48], Sedgewick [62] or Cormen et al [17].

These algorithms have several variations such as the *Median of Three Partition* for `QuickSelect` [47], the multikey `QuickSelect` [32], the approximate `QuickSelect` [51], the Bentley-Sedgewick variation for `QuickSort` [7], etc.

Why analyzing algorithms? The analysis of algorithms aims to analyze the main parameters of an algorithm in order to determine the amount of resources (time and storage) that are needed by the algorithm. The analysis of an algorithm provides a better understanding of the algorithm, may improve its implementation, helps predict its performance and its efficiency, and also allows to compare algorithms and classify them in difficulty.

Why focusing on average-case analysis? There are two main points of view when studying the performance of an algorithm: we may be interested in its worst-case performance or its average-case performance. Even if there exist some algorithms where the worst case

may be representative or “typical” of the observed behavior, more often, the worst-case rarely happens, and thus is a too pessimistic measure of the cost of the algorithm. This is why it is natural to deal with a probabilistic measure. The purpose of the average-case analysis of algorithms is to characterize the mean value of their “costs” under a well-defined probabilistic model that describes the initial distribution of the input.

Information theory. The efficiency of an algorithm which deals with words depends on two factors: the strategy of the algorithm itself and the mechanism, called the source, that generates the words constituting the input. We resort to information theory to define a general model of source that encompasses many interesting models such as memoryless sources (symbols are emitted independently), Markov chains (the emission of a symbol depends on the k previous symbols) and even more complicated sources with unbounded correlations between symbols. Unlike other random models which are usually uniform models, we consider in this thesis that the input is generated by a general source. And the model of source we choose encompasses other important models of sources (memoryless source, Markov chain, dynamical source). We adopt a parameterized model where each infinite word is characterized by a real number in the unit interval.

The probabilistic features of the source will play a crucial role in the analysis of the algorithm, when it deals with words emitted by the source, and there is indeed an interplay between the strategy of the algorithm and the probabilistic properties of the source. We wish to explain how the efficiency of the algorithm depends on the source. In particular, various notions of coincidence between words that will intervene in the analysis are characteristic of the studied algorithms. Our study is thus a tool for a better understanding of the algorithmic strategy.

Analytic combinatorics. There will be two phases in the analysis of the realistic cost of the algorithm, as it usual in analysis of algorithms. The first algebraic (or combinatorial) phase consists in computing the generating function (here a mixed Dirichlet series. The exact mean number of symbol comparisons performed by the algorithm is expressed as an alternative sum of the mixed Dirichlet series. This sum is usually complicated so that it is difficult to guess its behavior. The second analytic (or asymptotic) phase extracts the asymptotic behavior (which is generally not evident to guess at the end of the first phase) using a precise study of the generating function (seen as a function of the complex variable). The thesis deals with two main probabilistic models: the Poisson model where the number of words follows a Poisson law (where it is easier to work) and the Bernoulli model where this number is fixed (and is more natural in algorithmics). And it is possible to return to the Bernoulli model, with tools of complex analysis, like the Nörlund-Rice formula, the Mellin transform, the residue theorem, etc.

Contributions of the thesis.

In this thesis, we extend the applicability of the method which has been described in the paper [12]. We describe a new point of view on the basic algorithms, and their analysis, which can be (partially) automatized. Our study can be viewed as a first step towards the “realistic” analysis of sorting and searching algorithms, for two reasons: we consider a general model of inputs, and we deal with a (more) realistic cost, namely, the number of symbols which are needed to distinguish two words. This approach provides a fair comparison between comparison-based (e.g. **QuickSort**) and digital-based algorithms (e.g. **radix-sort** or trie-based algorithms) on a common ground.

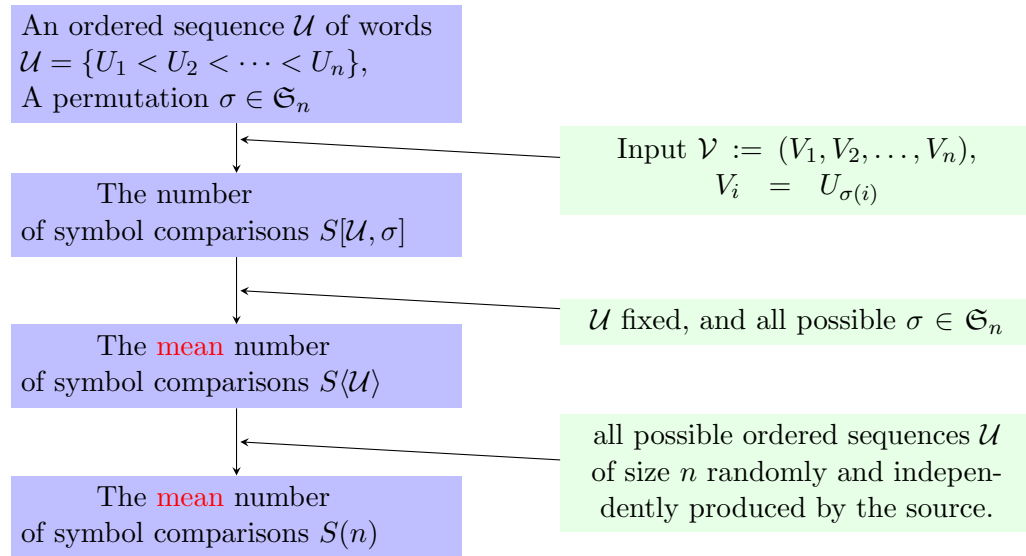


Figure 1: The steps for the analysis.

Methods. In our analysis, we always deal with two objects: an ordered set of infinite words $\mathcal{U} = \{U_1 < U_2 < \dots < U_n\}$ and an algorithm \mathcal{A} . For sorting and searching algorithms, which operate on words, the input is an array of n strings, each produced independently by a source. An input is described by a pair (\mathcal{U}, σ) where \mathcal{U} is a set of n words and σ a permutation of $[1..n]$.

The unit cost measure will no longer be the comparison between keys but between symbols. We denote by $S[\mathcal{U}, \sigma]$ the number of symbol comparisons performed by the algorithm \mathcal{A} on the set \mathcal{U} under the permutation σ . Varying over all possible $\sigma \in \mathfrak{S}_n$ (where n is the size of \mathcal{U}), we compute the mean number of symbol comparisons $S\langle \mathcal{U} \rangle$ which depends only on the set \mathcal{U} . In the next step, we fixed the cardinal n and consider all the possible sets \mathcal{U} generated by the source. Finally, the mean number of symbol comparisons $S(n)$ is the mean over all permutations $\sigma \in \mathfrak{S}_n$ and all possible sets of infinite words \mathcal{U} of size n randomly and independently produced by the source.

The general method is already described in [12], but this is a short paper, without much details. It was shown there that a “mixed” Dirichlet series denoted by $\varpi(s)$ characterizes the behavior of an algorithm with respect to the source. The thesis highlights the main principles of such an analysis, in order to make easier its application to various algorithms. As it is often the case in analytic combinatorics, there are two main phases in the method, a first phase where the series $\varpi(s)$ is built, and a second phase where it is analyzed with analytic tools. We show here how the first phase may mostly be performed in an “automatic” way.

Applications. We apply the method to three other popular algorithms: **InsertionSort**, **BubbleSort** and **SelectionMinimum**, respectively denoted in the sequel by the short names **InsSort**, **BubSort**, **SelMin** [14],[16] (see for instance the book [62] for a thorough description of these algorithms). With this approach, we also easily recover the results about the **QuickSort** and **QuickSelect** algorithms [15] that were already obtained in [12]. The main results are presented in Figure 2. The constants in the third column and the remainder terms depend on the source. The constants that depend on the algorithms will be explained respectively in Chapters 5,6,7. The remainder terms are precised in Theorem 6.2. The sub-

dominant terms will be explained in details in Chapters 5, 6, and 7. We then measure the robustness of the algorithms (i.e., the possible change in its complexity behavior, due to the change in the complexity measure, from the number of key comparisons to the number of symbol comparisons).

Algorithms	$K(n)$	Dominant term of $S(n)$	Subdominant terms of $S(n)$ (see Proposition 6.3)	Remainder term of $S(n)$ (see 6.2)
QuickSort	$2n \log n$	$\frac{1}{h(S)} n \log^2 n$	$\kappa_0 n \log n + \kappa_2 n$	$E(n)$
InsSort	$\frac{n^2}{4}$	$\frac{c(S)}{4} n^2$	$\frac{1}{h(S)} n \log n + \left(\kappa_0 - \frac{c(S)}{4} \right) n$	$E(n)$
BubSort	$\frac{n^2}{2}$	$\frac{1}{4h(S)} n^2 \log n$	$\left(\kappa_1 + \frac{c(S)}{4} \right) n^2$	$nE(n)$
QuickMin	$2n$	$2b(S) n$		$F(n)$
SelMin	n	$a(S) n$		$F(n)$

Figure 2: Results.

Alternative proofs. We provide also another alternative proof for the mean number of symbol comparisons of **QuickSort** and **InsSort**. This proof is obtained using the point of view given by Seidel in [65], where he relates the mean number of key comparisons and that of symbol comparisons (using the underlying structure of trie). In the case when the algorithm is strongly faithful (a property about the mean number of “local” comparisons between two keys by the algorithm), the mean number $S(n)$ may be deduced directly from the mean number $K(n)$.

A lower bound. The lower bound for the mean number of key comparisons is known to be $\Theta(n \log n)$ using the decision tree. We obtain a lower bound for the mean number of symbol comparisons, asymptotic to $\frac{1}{2 \log 2} \frac{1}{h(S)} n \log^2 n$. This shows that between comparison-based algorithms using lexicographic comparison of the symbol strings, **QuickSort** is always a quasi optimal sorting algorithm with respect to the number of symbol comparisons.

Thesis plan

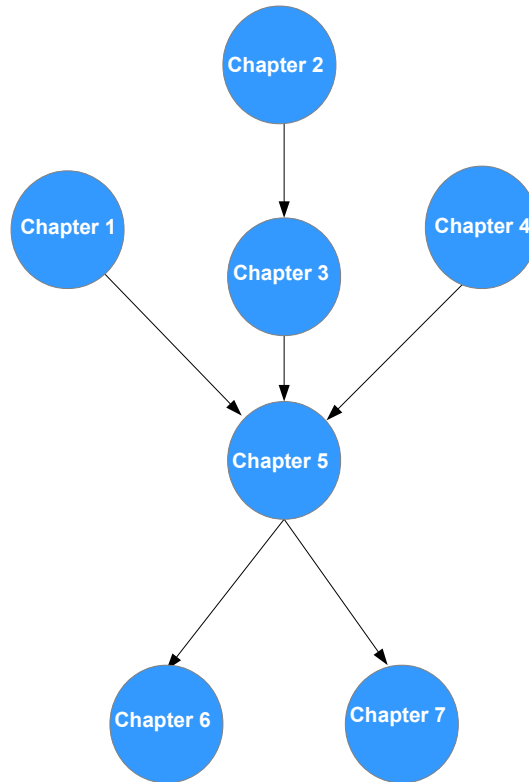


Figure 3: Plan of the thesis.

Chapter 1. In Chapter 1, we describe the strategy of the five studied algorithms; **QuickSort**, **QuickSelect**, **BubSort**, **InsSort** and **SelMin**. We also describe the permutation model that is used for the analysis of sorting and searching algorithms to get the mean number of key comparisons performed by these algorithms. Here, in this thesis, we are interested by a more precise cost, called $\pi_n(i, j)$, which is the mean number of “local” comparisons between the key of rank i and the key of rank j .

Chapter 2. Here, before analyzing an algorithm which now deals with words, we need to describe a model for the inputs, namely a model of source. We describe different models for simple sources and also a general model, which associates with each word a parameter of the unit interval. The correlation between emitted symbols can indeed be very low (in memoryless source, symbols are all independently generated), rely on a finite history (in the case of Markov chain sources, the emission of a symbol depends on the k previous symbols) but also possibly depend on an unbounded history (like in dynamical sources). We model the source with its fundamental probabilities which are the probabilities of emitting words

given their prefixes. The notion of coincidence is central in our study, and it is described by the “geometry” of the source, thanks to its fundamental triangles (defined in 2.19). Each fundamental triangle is associated to a prefix of word. It delimits a domain so that words of which the parameter belongs to this domain admit this prefix. The Dirichlet series of the source, denoted by $\Lambda(s)$, encapsulates many probabilistic properties of the source, and plays an important role here, as in many other studies.

Chapter 3. The probabilistic properties of the source depend on the analytic properties of its Dirichlet series $\Lambda(s)$. In this chapter, we study more deeply these properties. We distinguish two regions where we analyze the set of poles of $\Lambda(s)$: on the vertical line $\Re_s = 1$ and on the left half plane. The analysis is first done for simple sources (memoryless sources, Markov chains) and then extended to dynamical sources. We are in particular interested in “tameness” regions where $\Lambda(s)$ is meromorphic (with a good knowledge of its poles) and of polynomial growth when s becomes large. This will be essential for applying the Rice method (described in the next chapter). We recall conditions on fundamental probabilities that entail the series $\Lambda(s)$ to be tame, and relate the shape of the tameness region to specific properties of probabilities.

Chapter 4. Chapter 4 describes the main tools that will be used in our analysis: the Poisson model where the number of input items follows a Poisson law, and the Bernoulli model where this number is fixed. We describe the main tool which will be used in our analysis, the Rice method, which is related to the Mellin transform. Chapter 4 marks the end of the first part of the thesis. The next part describes the general framework for the realistic analysis and its application to sorting and searching algorithms.

Chapter 5. Chapter 5 draws a general framework for the realistic analysis of sorting and searching algorithms. The analysis has two main steps: the algebraic (or combinatorial) step and the analytic (or asymptotic) step. The first step aims at computing the exact mean number of symbol comparisons performed by the algorithm. We first work in the Poisson model, and we return to the Bernoulli model. The second step aims at deriving an asymptotic (and human readable) form of this exact formula, and we use the Rice method. While describing the general framework, we show its application in the analysis of **SelMin**.

Chapter 6. Chapter 6 applies the general framework 5 to three sorting algorithms, namely **QuickSort**, **InsSort**, **BubSort**. We discuss the robustness of the algorithm, i.e., the possible change in the complexity behavior, due to the change in the complexity measure, from the number of key comparisons to the number of symbol comparisons. In the second part of the Chapter, we provide an alternative approach for the analysis of **QuickSort** and **InsSort**. We use the point of view given by Seidel who relates the mean number of key comparisons and the mean number of symbol comparisons via the structure of a trie. In the case when the algorithm is strongly faithful (a property about the mean number of “local” comparisons between two keys, which is satisfied by both algorithms **QuickSort** and **InsSort**), the mean number $S(n)$ may be deduced directly from the mean number $K(n)$. This alternative proof does not apply to **BubSort** since it is not strongly faithful. The lower bound for the mean number of key comparisons of sorting algorithms is well-known to be $\Theta(n \log n)$ (using bounds on the size of a decision tree). We prove that the lower bound for the mean number of symbol comparisons for sorting algorithms using strings comparisons is $\frac{1}{2 \log 2h(S)} n \log^2 n$. This means that **QuickSort** is quasi-optimal in the model of symbol comparisons (as in the model of key comparisons).

Chapter 7. We analyze **QuickSelect** and its variations **QuickMin**, **QuickRand**, **QuickQuant** and **QuickVal** (a dual version of **QuickSelect** introduced in Chapter 7). The analysis of

QuickSelect was already presented in the paper [12] but several proofs remained implicit or have not been written yet. To get the asymptotics of the mean number of symbol comparisons of **QuickSelect**, we have to follow a slightly more complicated path (compared to previous sorting algorithms in Chapter 6 and the minimum selection that illustrates Chapter 5). We prove that its asymptotics is similar to that of the dual algorithm **QuickVal**, with the use of the Laplace method.

Relations with other works. This thesis was carried out in the GREYC Laboratory. During the same time, another PhD student in the GREYC, Kanak Hun, prepared his thesis [42] under the supervision of Brigitte Vallée, on the subject “Analysis of depth of digital trees built on general sources”. In fact, even if the subjects of the two theses are different, they share two main important objects: general sources and Poisson-Rice methodology. This is why there exist Chapters in our theses which are very closely related: The present Chapter 2 shares many sections with Chapter 2 of Kanak’s thesis, and our Chapter 4 is almost the same as Chapter 4 of Kanak’s thesis.

During the preparation of the thesis, Eda Cesaratto and Brigitte Vallée wrote the last version of their paper [11] and we worked together for the precise proof of Proposition 4.9 which finally appeared in the paper [11], in the present thesis and also in Kanak’s thesis [42].

Chapter 1

Sorting and searching algorithms. Analyses in the key model.

Contents

1.1	Algorithms and analysis of algorithms.	10
1.1.1	A general introduction	10
1.1.2	Sorting and searching algorithms: the input key model.	11
1.1.3	Sorting and searching algorithms: the costs of interest.	12
1.1.4	Important parameters of permutations.	13
1.1.5	A lower bound for sorting in the key model	14
1.1.6	The main algorithms under study.	16
1.2	The QuickSort algorithm.	16
1.2.1	Description	16
1.2.2	Mean number of key-comparisons.	17
1.2.3	Mean number of local comparaisons.	19
1.3	The QuickSelect algorithm and its variants	20
1.3.1	Description	20
1.3.2	Mean number of key comparisons.	20
1.3.3	Mean number of local key-comparisons.	24
1.4	The InsertionSort algorithm.	25
1.4.1	Description	25
1.4.2	Mean number of key comparaisons	26
1.4.3	Mean number of local key comparisons.	27
1.5	The BubbleSort algorithm	27
1.5.1	Description.	27
1.5.2	Mean number of comparisons.	28
1.5.3	Mean number of local key comparisons.	29
1.6	The Selection-Minimum Algorithm	30
1.7	Towards a realistic analysis of selection and sorting algorithms.	31
1.7.1	Conclusion on the mean number of local comparisons.	31
1.7.2	Normalized and non normalized algorithms.	31

In this chapter, we first describe the general principles in analysis of algorithms, and give more details in the case of sorting and searching algorithms, where the input is a sequence of keys, and the main parameter is the number of key comparisons. We focus on the average-case analysis of two main parameters. We recall the classical results on the total number of key-comparisons, but we also study a “new” parameter (much less studied) which plays a central role in the present thesis: the number of “local” comparisons, namely comparisons between the key of rank i and the key of rank j .

We then present, in the following sections, the five algorithms which will be the main actors of the thesis: these are the popular algorithms **QuickSort**, **QuickSelect**, and its variants **QuickMin** and **QuickRand**, together with *Insertion Sort* (denoted here as **InsSort**), *Bubble Sort* (denoted here as **BubSort** and finally *Minimum Selection* (denoted as **SelMin**). For each algorithm, we focus on the average-case analysis of our two main parameters in the so-called permutation model. In particular, we perform a precise analysis of the new parameter (the number of “local” comparisons), which also allows us to recover (often very easily) the expression of the mean number of total comparisons.

1.1 Algorithms and analysis of algorithms.

1.1.1 A general introduction

An algorithm is a mapping $\mathcal{A} : \Omega \rightarrow S$. The set Ω is the set of inputs and the set S is the set of outputs. An algorithm is often described as a sequence of elementary steps.

The set of inputs. The first task is to choose a model for the inputs, with a good notion of size. The size, denoted by $|\cdot|$, is a mapping $|\cdot| : \Omega \rightarrow \mathbb{N}$. The size measures the space which is occupied by the input. This is not an exact measure. For an input ω , the size $|\omega|$ is roughly the space which is needed to represent the input ω (or more rigorously is linearly related to this space). For any $n \in \mathbb{N}$, the set Ω_n gathers the inputs of size n , namely

$$\Omega_n := \{\omega \in \Omega; \quad |\omega| = n\}.$$

Parameters of interest. The algorithm \mathcal{A} is often a very complex sequence of operations, and its exact description is not easy. In fact, all the operations performed have not the same importance, and one isolates the main operations performed to better analyze them. Very often, one first defines what is the main basic operation of interest, and its cost, and the (total) cost of interest is the sum of the costs of all the basic operations performed by the algorithm. Such a cost c depends on the input ω , and is then a function $c : \Omega \rightarrow \mathbb{N}$ which associates to the input ω the cost $c(\omega)$ performed by the algorithm \mathcal{A} on the input ω .

Analysis of algorithms. This sub-domain of the algorithmics domain aims to analyze the main parameters of an algorithm \mathcal{A} , in order to determine the amount of resources (time and storage) that are needed by the algorithm \mathcal{A} . Analysis of algorithms provides a better understanding of algorithms, improves the implementation, predicts the performance and the efficiency of algorithms, allows to compare algorithms and classify them in difficulty.

For this purpose, it is natural to analyze the main parameters of an algorithm \mathcal{A} , as a function of the input size. When the set of inputs Ω is modelled, when the size $|\cdot|$ is chosen,

and when the cost c of interest is isolated, we consider the restriction of c to the set Ω_n of the inputs of size n , and, in particular, the image

$$c(\Omega_n) := \{c(\omega); \omega \in \Omega_n\},$$

and we wish to describe its behaviour in an asymptotic way, when the size n tends to ∞ . We may be interested by the minimum of $c(\Omega_n)$, or the maximum of $c(\Omega_n)$. This gives rise to the best-case $\underline{C}(n)$, or the worst-case $\widehat{C}(n)$, namely,

$$\underline{C}(n) = \min\{c(\omega), \omega \in \Omega_n\}, \quad \widehat{C}(n) = \max\{c(\omega), \omega \in \Omega_n\}.$$

However, these two measures are very often rough measures, since they are not “typical”. For some algorithms, the worst case (and the best case) happens frequently, and they are in a sense “typical”, but, for many other algorithms, the worst-case (for instance) rarely happens, and this is a too pessimistic measure of the cost c .

Probabilistic analysis of algorithms. This is why it is natural to deal with a probabilistic measure. One first chooses a probabilistic distribution \mathbb{P}_n on each subset Ω_n . This choice is not easy to do, since this distribution must be both realistic (with respect to the usual application of the algorithm), and yet manageable (as a probabilistic tool). Then, the algorithm \mathcal{A} and the cost c , or more precisely their restrictions to the set Ω_n , become random variables on the set Ω_n . The probabilistic analysis of the algorithm \mathcal{A} studies these random variables. As it is usual in probabilistic contexts, one begins by the expectation,

$$C(n) := \mathbb{E}_n[c] = \int_{\Omega_n} c(\omega) d\mathbb{P}_n(\omega),$$

but we may be interested by other measures, as the variance of c_n , or the limit distribution of the variable c_n . When we limit ourselves to the study of the mean of the cost c_n , we perform average-case analysis of the algorithm \mathcal{A} . This is the general framework of this thesis.

Even when we restrict ourselves to the average-case analysis, it is often not easy to obtain an exact formula for the mean value of the cost. And, when it is the case, it may not be clear whether this exact formula provides a precise knowledge of the behaviour of the algorithm when $n \rightarrow \infty$. This is the case of the studies of the thesis, where we often obtain exact formulae which are not easy to read. This is why the final description is:

The average-case analysis of algorithms aims to provide asymptotic estimates for the mean values of the main parameters of interest, when the input size n becomes large.

This explains why we will deal with the classical notions widely used in asymptotic studies as O , o , Θ and Ω , which are recalled in the annex.

1.1.2 Sorting and searching algorithms: the input key model.

The initial model. Consider any (totally) ordered set \mathcal{O} , and the Cartesian product \mathcal{O}^n .

A *selection algorithm* \mathcal{A} is a family of mappings \mathcal{A}_n , where the mapping $\mathcal{A}_n : \mathcal{O}^n \rightarrow \mathcal{O}^n$ computes, from a sequence (V_1, V_2, \dots, V_n) , the ordered sequence (U_1, U_2, \dots, U_n) (in the increasing order) built from the initial sequence. If the initial sequence is formed with

distinct elements, there exists a (unique) permutation $\sigma \in \mathfrak{S}_n$ for which $V_i = U_{\sigma(i)}$, and the task of the sorting algorithm is to build this (unknown) permutation. This permutation is called the underlying permutation.

A *searching algorithm* is a family of mappings \mathcal{A}_n , where the mapping $\mathcal{A}_n : \mathcal{O}^n \times [1..n] \rightarrow \mathcal{O}_n$ outputs, on the input formed with a sequence (V_1, V_2, \dots, V_n) and an index $m \in [1..n]$, the element V_i whose rank equals m : This means that $V_i = U_m$, and the searching algorithm outputs in fact the index i on which the underlying permutation σ satisfies $\sigma(i) = m$. When $m = 1$, the algorithm outputs the minimum; when $m = n$, the algorithm outputs the maximum; when $m = \lfloor n/2 \rfloor$, the algorithm computes the median.

In classical algorithmic studies, what is the ordered set \mathcal{O} ? In fact, this is not really defined; it is just said: “this is a set of keys”. And the usual definition of a key is

A key is a data fragment whose internal structure is not taken into account.

Since the internal structure of a key is not specified, it is natural to give it a unit size. Then any input of \mathcal{O}^n is given the size n .

In this context, there are two natural operations on the set \mathcal{O}^n —comparisons and exchanges—, and we assume that the algorithm \mathcal{A} (either a sorting or a searching algorithm) uses only these two basic operations.

The permutation model. In fact, for sorting and searching algorithms, as defined above, the set \mathcal{O} has no clear meaning and no influence on the behaviour of the algorithm, that only depends on the underlying permutation, as we now explain:

Consider, for some fixed n , two inputs of \mathcal{O}^n , namely two sequences (V_1, V_2, \dots, V_n) and $(V'_1, V'_2, \dots, V'_n)$. Denote by (U_1, U_2, \dots, U_n) and $(U'_1, U'_2, \dots, U'_n)$ the two ordered sequences, and consider the two underlying permutations $\sigma, \sigma' \in \mathfrak{S}_n$ for which, for any $i \in [1..n]$, one has $V_i = U_{\sigma(i)}$ and $V'_i = U'_{\sigma'(i)}$. Assume that the equality $\sigma = \sigma'$ holds. Then the algorithm \mathcal{A} performs exactly the same way on the two inputs (V_1, V_2, \dots, V_n) and $(V'_1, V'_2, \dots, V'_n)$. This is why it is natural to define an equivalence relation \sim_n on \mathcal{O}^n ,

$$(V_1, V_2, \dots, V_n) \sim_n (V'_1, V'_2, \dots, V'_n) \iff \text{the underlying permutations } \sigma \text{ and } \sigma' \text{ are equal.}$$

Now, each equivalence class is represented by a permutation $\sigma \in \mathfrak{S}_n$, and the quotient \mathcal{O}^n / \sim_n is in bijection with the set \mathfrak{S}_n of permutations. Then, the set \mathcal{O}^n can be replaced by the set \mathfrak{S}_n of permutations over $[1..n]$, endowed with the uniform probability. This is the permutation model, where the notion of key is completely forgotten.

Here, we use an intermediate model, which will be easier to modify in the next steps of our study. We consider as possible inputs of size n the pairs (\mathcal{U}, σ) formed with a fixed ordered sequence of keys $\mathcal{U} = \{U_1 < U_2 < \dots < U_n\}$ together with a variable permutation $\sigma \in \mathfrak{S}_n$. The set of inputs of size n is then equal to $\{\mathcal{U}\} \times \mathfrak{S}_n$. It is in bijection with \mathfrak{S}_n and it is endowed with the uniform probability: this means that all the sequences $\mathcal{V} = (V_1, V_2, \dots, V_n)$ obtained by the action of a permutation σ on \mathcal{U} via the equalities $V_i = U_{\sigma(i)}$ are then equiprobable. Note that, on all these input sequences, the sorting algorithm always outputs the sequence \mathcal{U} . Note that later on along the study, the sequence \mathcal{U} will become a variable sequence of words.

1.1.3 Sorting and searching algorithms: the costs of interest.

Very often, and this is the case in this thesis, we focus on the comparison operation: this will be the *basic operation* of the algorithm, with a unit cost. More precisely, for each algorithm (either a sorting or searching algorithm), we consider two types of comparisons.

Total number of comparisons. The usual cost of interest is the total number of comparisons performed by the algorithm \mathcal{A} on the input (\mathcal{U}, σ) . The mean number of comparisons is the expectation of this random variable when \mathfrak{S}_n is endowed with the uniform probability. It is denoted by $K(n)$.

Number of “local” comparisons, between keys U_i and U_j . The previous cost is a “global” cost, and it proves useful to consider also “local” costs, which provide a more precise knowledge about the behaviour of the algorithm. Namely, for any pair (i, j) with $1 \leq i < j \leq n$, we consider the number of comparisons performed by the algorithm \mathcal{A} between the pair (U_i, U_j) of keys, namely the key U_i of rank i and the key U_j of rank j , and we denote its mean by $\pi_n(i, j)$.

There are two possible types of comparisons between two keys U_i and U_j :

- the *positive* comparisons which occur when U_i and U_j arrive in the good order: this is the case, when, in the initial sequence, and for $i < j$, the key U_i arrives before the key U_j .
- the *negative* comparisons which occur when U_i and U_j (for $i < j$) arrive in the wrong order, namely U_i arrives after U_j .

The mean number of positive and negative comparisons between two keys U_i and U_j is denoted respectively by $\pi_n^+(i, j)$ and $\pi_n^-(i, j)$. The mean number of key comparisons is $\pi_n(i, j) = \pi_n^+(i, j) + \pi_n^-(i, j)$. And the equality

$$K(n) = \sum_{1 \leq i < j \leq n} \pi_n(i, j)$$

is often an easy tool for recovering the classical formula for $K(n)$.

1.1.4 Important parameters of permutations.

We repeat that we study random variables defined on the set \mathfrak{S}_n . We recall three important variables: arrival times, inversions and left-to-right minimum.

Arrival times. When the ordered sequence \mathcal{U} is given under the permutation σ , the input is the sequence (V_1, V_2, \dots, V_n) , with $V_i = U_{\sigma(i)}$. It is also denoted by the pair (\mathcal{U}, σ) . It is often more intuitive to deal with the arrival times. The *arrival time* of U_i , denoted by $\tau(i)$ is the position j of U_i in the input array. We will also use the convenient notation $\mathcal{U}_{[i, j]}$ which gathers all keys whose ranks are in $[i, j]$.

Example. Consider the ordered sequence with three elements $\mathcal{U} = (-2.02, \sqrt{2} \approx 1.44, 100)$ and $\mathcal{V} = (100, -2.02, \sqrt{2})$. The permutation $\sigma = (312)$ acts on the ordered set \mathcal{U} and yields the sequence \mathcal{V} . The arrival times are given by $\tau = (231)$.

Of course, since the sequence \mathcal{U} is fixed, there is a simple relation between the two points of view: the equality $\tau(i) = j$ means $V_j = U_i$ and also $\sigma(j) = i$; thus, there is a bijection between the vector τ of arrival times and the underlying permutation σ , and τ is associated to the inverse permutation σ^{-1} of σ . Then, for a fixed index i , the arrival time $\tau(i)$ is a mapping $[1 \dots n] \rightarrow [1 \dots n]$ which defines a uniform random variable, and, for two distinct indices i, j , the two random variables $\tau(i)$ and $\tau(j)$ are independent. Then one has

$$\Pr_n[\tau(i) = k] = \frac{1}{n}, \quad \Pr_n[\tau(i) = k \text{ and } \tau(j) = \ell] = \frac{1}{n^2}.$$

The mean numbers $\pi_n^\pm(i, j)$ defined in the previous subsection are often computed in a separate way, with direct probabilistic arguments dealing with the arrival times. We will see that the event “ U_i and U_j are compared” is generally “similar” to an event of the type “*The arrival times of the keys U_i and U_j into a given subset \mathcal{W} of keys are the first two (resp. the last two)*”. For a subset \mathcal{W} of cardinal ℓ , the probability of such an event is $\frac{1}{\ell(\ell-1)}$. Moreover, the subset \mathcal{W} is often a subset $\mathcal{U}_{[x,y]}$ which gathers all the keys of the set \mathcal{U} whose rank belongs to the interval $[x, y]$, with three main cases, according to the algorithms: $[x, y] = [1, i]$, $[x, y] = [1, j]$, or $[x, y] = [i, j]$, which entails that ℓ belongs to $\{i, j, j - i + 1\}$. This explains why the expectations $\pi_n^\pm(i, j)$ are usually expressed as sums of rational functions depending on i, j or $j - i$.

Inversions. An inversion of a permutation σ is a pair of indices (i, j) with $1 \leq i < j \leq n$ for which $\sigma(i) > \sigma(j)$ (the images are in the reverse order). Using the Iverson bracket $\llbracket \cdot \rrbracket$ (see [36]), we define,

$$I_{(i,j)}(\sigma) := \llbracket \sigma(i) > \sigma(j) \rrbracket, \quad \text{for a pair } (i, j) \text{ with } i < j, \quad I(\sigma) := \sum_{1 \leq i < j \leq n} I_{(i,j)}(\sigma).$$

Then, when \mathfrak{S}_n is endowed with the uniform probability, the equalities hold

$$\mathbb{P}_n[I_{(i,j)} = 1] = \frac{1}{2} \quad \mathbb{E}_n[I] = \sum_{1 \leq i < j \leq n} \frac{1}{2} = \frac{1}{4}n(n-1).$$

Left to right minima. A left-to-right minimum is an index j for which all index $k < j$ the inequality $\sigma(k) > \sigma(j)$ holds. With the point of view of arrival times, a left to right minimum is an element U_j which arrives before all the keys of $\mathcal{U}_{[1,j]}$. This is the first-in key of the subset $\mathcal{U}_{[1,j]}$. The probability that U_j be a left-to-right minimum is equal to $1/j$. The mean number of left-to-right minima equals the harmonic number

$$H_n = 1 + \frac{1}{2} + \cdots + \frac{1}{n}. \tag{1.1}$$

We will see that the harmonic number plays a central role in our average-case analyses.

1.1.5 A lower bound for sorting in the key model

We assume again that all the elements to be sorted are distinct. Then tests of the form “ $V_i = V_j$ ” are useless, as we can assume that no tests of this form are made. Then the algorithm only performs the tests denoted “ $V_i : V_j$ ” which have only two possible outputs $V_i > V_j$ or $V_j > V_i$.

We associate with a sorting algorithm \mathcal{A} its decision tree. This is a binary tree which represents the comparisons performed by this algorithm when it operates on an input of a given size n . Flow of control, data movement, and all other aspects of the algorithm are ignored. Each internal node, labelled by $i : j$, with $1 \leq i < j \leq n$ represents a comparison between the two keys V_i and V_j that are at positions i and j at the beginning of the algorithm. Any decision node has (at most) two subtrees: the left subtree dictates subsequent comparisons when the inequality $V_i < V_j$ holds and the right subtree dictates subsequent comparisons when the inequality $V_i > V_j$ holds. For a clever algorithm, each internal node has exactly two subtrees, but there are algorithms for which some internal nodes have only one subtree. Each leaf is labelled by the permutation τ of the arrival times, represented by the sequence $(\tau(1), \tau(2), \dots, \tau(n))$. The execution of the algorithm

\mathcal{A} on an input with distinct elements corresponds to tracing a path from the root to one leaf, and the path is completely determined by the permutation τ of arrival times, which is the inverse of the permutation σ which underlies the input.

For instance, Figure 1.1 represents the decision tree of Insertion Sort when operating on three elements. The bold path represents the execution of the algorithm on the input (V_1, V_2, V_3) when it satisfies $V_3 < V_1 < V_2$. This means that $V_3 = U_1, V_1 = U_2, V_2 = U_3$: The arrival times are $\tau(U_1) = 3, \tau(U_2) = 1, \tau(U_3) = 2$, and the underlying permutation is $\sigma = \tau^{-1}$ and satisfies $\sigma(1) = 2, \sigma(2) = 3, \sigma(3) = 1$.

There is an (easy and) important result:

Lemma 1.1. *A decision tree for a sorting algorithm \mathcal{A} when it operates on n keys has exactly $n!$ leaves, each of them labelling a distinct underlying permutation.*

Proof. On each input (\mathcal{U}, σ) , the execution of the algorithm takes a path which finishes at the leaf labelled by τ . It is not possible to have two distinct leaves which labels the same permutation: this would contradict the fact that we deal with deterministic algorithms. \square

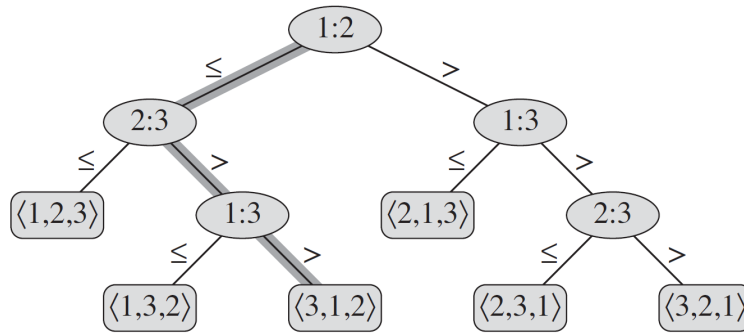


Figure 1.1: Decision tree of the Insertion Sort Algorithm when operating on three elements

The length of the longest path from the root to any of its leaves represents the worst-case number of comparisons that the corresponding sorting algorithm performs. Hence, the worst-case number of comparisons for a comparison sort equals the height H of its decision tree. In the same vein, the average-case number of comparisons for a comparison sort equals the mean length of a branch (called the depth D) of its decision tree (at least in the uniform model of permutations).

For a binary tree, the height H , the depth D , and the number of leaves k satisfy the following inequalities

$$k \leq 2^D \leq 2^H.$$

Since a decision tree in size n has exactly $n!$ leaves, this shows the two inequalities

$$H \geq D \geq \log_2(n!).$$

The Stirling formula $n! \sim n^n e^{-n} \sqrt{2\pi n}$ proves the estimate $\log_2 n! = \Omega(n \log n)$. We have obtained a lower bound for the number of comparisons in the worst-case, and in the average-case.

Proposition 1.2. *Any sorting algorithm which only performs comparisons and exchanges performs at least a number of comparisons which is in the worst-case and in the average case $\Omega(n \log n)$. ([17])*

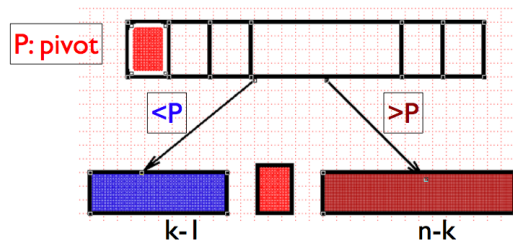
1.1.6 The main algorithms under study.

In the thesis, we mainly study five algorithms: three sorting algorithms —QuickSort, Insertion Sort and Bubble Sort—and two selection algorithms —QuickSelect and Selection Minimum. The description of these algorithms can be found in the textbooks [49, 1, 17, 9]. Here, in this first chapter, we describe these algorithms, and recall the well-known results about their average-case complexity, denoted as $K(n)$, when the cost is the total number of key-comparisons. But, as we already mentioned it in Section 1.1.3, we are also interested (for reasons which will appear more precisely at the end of this chapter) in another cost, the number of comparisons between the key of rang i and the key of rank j , and by its mean, denoted by $\pi_n(i, j)$. This type of average-case complexity is not classically studied and the expressions of $\pi_n(i, j)$ for each of the five algorithms of interest appear to be new. They are summarized at the end of this chapter in the table of Figure 1.12.

1.2 The QuickSort algorithm.

1.2.1 Description

The algorithm was introduced in 1962 by C.A.R Hoare in his original papers [38, 39, 41] and be improved by several authors as Knuth [50], van Emden [73], Sedgewick [59]. It has been cited [22] as one of the ten algorithms “with the greatest influence on the development and practice of science and the engineering in the 20th century” . This algorithm is based on the “Divide and Conquer” principle. All the keys are compared to the first key of the array that is used as a “pivot”. During the **Partition** stage, each key is compared to the pivot, and the keys that are smaller than the pivot are placed on its left in the array, whereas the keys that are greater are placed on its right. After this partitioning process, the pivot is at the right place, and we know its rank. Then, the algorithm recursively sorts the two sub-arrays, the left subarray and the right subarray. In particular, for the problem of equal keys in the array, one can use the tenary partition [60][74][6]. The problem of how implementing this algorithm in an effective way is presented in [61]. The multikey QuickSort is another variant of the regular QuickSort algorithm [7].



They are various possibilities for implementing the **Partition** process. It is first important to remark that all the partition processes share the common following features: each element of the array $V[\text{left} \dots \text{right}]$ with $i \neq \text{left}$ is compared to the pivot $V[\text{left}]$ (and thus the number of comparisons is always greater or equal to $n - 1$). At the end of the procedure, the pivot is at the right place, the keys that are smaller than the pivot are placed on its left, and keys that are greater than the pivot are placed on its right. However, the number of exchanges, and the final positions of the elements inside the two sub-arrays, the left one and the right, may depend on the precise implementation of the **Partition** procedure.

We now describe the implementation described in Figure 1.2, see also [64]. We choose the first element of the array $V[\text{left} \dots \text{right}]$ as the pivot, and the variable v holds the value of the pivot $V[\text{left}]$. We use two scan pointers i and j , the left pointer i scans from the

left, and the right pointer j scans from the right. Each time we find a pair (i, j) such that $V[i] \geq v$ and $V[j] \leq v$, we exchange $V[i]$ and $V[j]$. When the pointers cross each other, an extra exchange of $V[i]$ and $V[j]$ is done with $j < i$ just after the pointers cross (but before the crossing is detected) and the outer **repeat** loop exited. The last three assignments implement the exchange between $V[i]$ and $V[j]$ (to undo the extra exchange), and between $V[i]$ and $V[\text{left}]$ to put the pivot into position.

Procedure Partition($V, \text{left}, \text{right}$)

```

/* Partition (V, left, right) rearranges the sub-array V[left..right] according to
   its first element V[left], called the pivot, and returns the position of
   the pivot after partitioning */
v ← V[left] ;
i ← left;
j ← right;
repeat
    repeat i ← i + 1 until V[i] ≥ v;
    repeat j ← j - 1 until V[j] ≤ v;
    t ← V[i]; V[i] ← V[j]; V[j] ← t;
until j ≤ i;
t ← V[i]; V[i] ← V[left]; V[left] ← t;

```

Figure 1.2: Partition Algorithm

Procedure Quicksort($V, \text{left}, \text{right}$)

```

/* Sorts the sub-array V[left..right]. */
/* Recursive function to be called for an array V[1..n]: Quicksort (V, 1, n) */
if left < right then
    k ← Partition(V, left, right)
    Quicksort (V, left, k - 1)
    Quicksort (V, k + 1, right)
end

```

Figure 1.3: QuickSort Algorithm

1.2.2 Mean number of key-comparisons.

The method for analyzing **QuickSort** is typical in the analysis of a broad class of recursive algorithms, or recursive data structures which are central in algorithmics.

Lemma 1.3. *The mean number of key comparisons performed by **QuickSort** on a random input of size n involves the harmonic number H_n under the form*

$$K(n) = 2(n+1)H_n - 4n \sim 2n \log n, \quad \text{with} \quad H_n = \sum_{i=1}^n \frac{1}{i}.$$

The maximum number of key comparisons satisfies

$$\widehat{K}(n) = \frac{n(n-1)}{2} \sim \frac{n^2}{2}.$$

Proof. We consider the two cases (worst-case, then average case).

Worst case. We denote by $\widehat{K}(n)$ the maximal number of keys comparisons. By convention, one has $\widehat{K}(0) = \widehat{K}(1) = 0$. For $n \geq 2$, the worst-case arises when the keys are already sorted in increasing order. In this case, the first key (chosen as the pivot) is the minimum, and, after the first partition stage which performs $n - 1$ comparisons, the left sub-array is empty whereas the right sub-array contains the other $n - 1$ keys, which are also sorted in increasing order. Then, the following recurrence holds on the sequence $\widehat{K}(n)$,

$$\widehat{K}(n) = \widehat{K}(n-1) + n - 1, \quad \text{and then} \quad \widehat{K}(n) = 1 + 2 + \cdots + (n-1) = \frac{n(n-1)}{2}.$$

Average case. The probability that the pivot has the rank $i + 1$ equals $1/n$ for any $i \in [0 \dots n-1]$. In the case where the pivot has rank $i + 1$, the left sub-array has i elements, and the right sub-array has $n - i - 1$ elements. Moreover, if the initial array is random, it is the same for the two sub-arrays. Dealing with conditional expectations, we obtain a recurrence on the sequence $K(n)$:

$$K(n) = n - 1 + \frac{1}{n} \sum_{i=0}^{n-1} (K(i) + K(n-i-1)) = (n-1) + \frac{2}{n} \sum_{i=0}^{n-1} K(i).$$

This recurrence depends on the whole previous history, since each $K(n)$ is defined in terms of all the previous terms $K(i)$ for $i < n$. To solve this recurrence, we begin with the two relations (obtained by multiplying by n and shifting)

$$nK(n) = n(n-1) + 2 \sum_{i=0}^{n-1} K(i), \quad (n-1)K(n-1) = (n-1)(n-2) + 2 \sum_{i=0}^{n-2} K(i).$$

With a subtraction,

$$nK(n) - (n-1)K(n-1) = 2K(n-1) + 2(n-1), \quad \text{and then} \quad nK(n) = (n+1)K(n-1) + 2(n-1).$$

We now define a new sequence as $t(n) := K(n)/(n+1)$, which satisfies

$$t(n) = t(n-1) + \frac{2(n-1)}{n(n+1)} = t(n-1) - \frac{2}{n} + \frac{4}{n+1}.$$

Finally,

$$t(n) = 2H_n + \frac{4}{n+1} - 4, \quad \text{and thus} \quad K(n) = 2(n+1)H_n - 4n.$$

As the harmonic number H_n satisfies $H_n \sim \log n$, the asymptotic behavior of $K(n)$ is

$$K(n) \sim 2n \log n \quad \text{or} \quad K(n) \sim 2n \log n \sim 1.39 n \log_2 n.$$

Alternative proof using generating functions. We can use the generating function $G(z)$ of the sequence $K(n)$ to obtain the exact expression of $K(n)$. Beginning with the expression of $G(z)$ and the recurrence on the sequence K_n

$$G(z) := \sum_{n \geq 0} K(n)z^n, \quad nK(n) = n(n-1) + 2 \sum_{i=0}^{n-1} K(i), \quad K(0) = K(1) = 0,$$

we obtain

$$\sum_{n \geq 1} nK(n)z^n = \sum_{n \geq 1} n(n-1)z^n + 2 \sum_{n \geq 1} \sum_{i=0}^{n-1} K(i)z^n.$$

The left-side term is $zG'(z)$, and the two generating functions terms on the right side are

$$\frac{2z^2}{(1-z)^3}, \quad \text{and} \quad 2\frac{zG(z)}{1-z}.$$

Then the generating function $G(z)$ satisfies a linear differential equation of the form

$$G'(z) = \frac{2z}{(1-z)^3} + \frac{2G(z)}{1-z}, \quad \text{whose homogeneous version is} \quad G'(z) = \frac{2G(z)}{1-z}.$$

The homogeneous equation admits the solution $G_1(z) = 1/(1-z)^2$. We now look for a particular solution of the complete equation under the form $G_1(z)G_0(z)$. The function $G_0(z)$ satisfies

$$G'_0(z) = \frac{2z}{1-z}, \quad \text{so that} \quad G_0(z) = -2\log(1-z) - 2z, \quad \text{and} \quad G(z) = \frac{-2\log(1-z) - 2z}{(1-z)^2}.$$

The following expansions,

$$\frac{1}{(1-z)^2} = \sum_{n=0}^{\infty} (n+1)z^n, \quad \log \frac{1}{1-z} = \sum_{n=1}^{\infty} \frac{z^n}{n}$$

lead to the expressions

$$\begin{aligned} \frac{1}{(1-z)^2} \log \frac{1}{1-z} &= \sum_{n=1}^{\infty} \left[\sum_{k=1}^n \frac{n+1-k}{k} \right] z^n = \sum_{n=1}^{\infty} \sum_{k=1}^n \left[\frac{n+1}{k} - 1 \right] z^n, \\ [z^n] \frac{-1}{(1-z)^2} \log(1-z) &= (n+1)H_n - n \quad \text{and} \quad [z^n] \frac{z}{(1-z)^2} = n. \end{aligned}$$

Finally, the coefficient $K(n)$ of the generating function $G(z)$ satisfies

$$\begin{aligned} K(n) &= [z^n] \left(\frac{-2}{(1-z)^2} \log(1-z) - \frac{2z}{(1-z)^2} \right) \\ &= 2(n+1)H_n - 4n. \end{aligned}$$

This is the same result as above! □

The generating function together with the symbolic method A.1.2 are also used to study the probability distribution of several characteristic parameters (average value, higher moments of the distribution of the cost function, both exactly and asymptotically) on various forms of QuickSort such as the “median of k” or “cutting small list” [37].

1.2.3 Mean number of local comparisons.

Lemma 1.4. *The mean number of key-comparisons performed by QuickSort algorithm between the key of rank i and the key of rank j ($1 \leq i < j \leq n$) is*

$$\pi_n(i, j) = \frac{2}{j-i+1}.$$

Proof. While the pivot does not belong to the subset $\mathcal{U}_{[i,j]}$, this set is not separated by the pivot, and the keys U_i and U_j are not compared. When the pivot belongs to the subset $\mathcal{U}_{[i,j]}$, the keys U_i and U_j may be compared only if U_i or U_j is a pivot. This event coincides with the event “ U_i or U_j is the first key-in inside the subset $\mathcal{U}_{[i,j]}$ ”. After such a comparison, the keys are separated and no longer compared. In the permutation model, the probability of the event “ U_i or U_j is the first key-in inside the subset $\mathcal{U}_{[i,j]}$ ” equals $2/(j-i+1)$, which leads to the result. □

It is now easy to recover the expression of the mean number $K(n)$ of comparisons between all the keys. It is expressed with the sum of harmonic numbers H_i under the form

$$K(n) = \sum_{1 \leq i < j \leq n} \frac{2}{j-i+1} = 2 \sum_{j=1}^n \sum_{i=1}^{j-1} \frac{1}{j-i+1} = 2 \sum_{j=1}^n \sum_{k=2}^j \frac{1}{k} = 2 \sum_{j=1}^n (H_j - 1).$$

The equality

$$\sum_{i=1}^n H_i = n + \frac{n-1}{2} + \dots + \frac{n-(j-1)}{j} + \dots + \frac{1}{n} = (n+1)H_n - n$$

then leads to the expression of $K(n)$.

1.3 The QuickSelect algorithm and its variants

1.3.1 Description

On an array formed with n (distinct) keys, the algorithm **QuickSelect**(m, n) returns the key of rank m . The **QuickSelect** algorithm is often called Hoare's Find algorithm [40]. As the **QuickSort** algorithm, it is based on the "Divide and Conquer" principle. It also uses the first key of the array as a pivot and performs the **Partition** operation. However, unlike **QuickSort**, after each **Partition** stage, the algorithm **QuickSelect** continues only on one of the two sub-arrays, the subarray which contains the key of rank m . An extension of the basic algorithm is the median-of-three Selection due to Singleton [66] in which the pivot key is chosen as the median of the first, middle and the last keys of the set. Anderson and Brown also analyze the combinatorial aspects of the algorithm in [2].

The algorithm **QuickSelect**(m, n) has many variants, depending on the value of the rank m .

- (i) The algorithms **QuickMin** and **QuickMax** are particular cases of **QuickSelect**(m, n) when $m = 1$ and $m = n$ respectively. The algorithm **QuickMin** returns the minimum key of the array, whereas the algorithm **QuickMax** returns the maximum key.
- (ii) The algorithm **QuickQuant** $_{\alpha}$ returns the α -quantile, namely the key of rank $\lfloor 1 + \alpha(n-1) \rfloor$. With this definition, the algorithm **QuickQuant** $_0$ corresponds to **QuickMin** and the algorithm **QuickQuant** $_1$ corresponds to **QuickMax**.
- (iii) The algorithm **QuickMed** returns the median key ($m = \lfloor \frac{n+1}{2} \rfloor$) of the array.
- (iv) The algorithm **QuickRand** outputs a key whose rank is uniformly chosen in the interval $[1..n]$.

1.3.2 Mean number of key comparisons.

The κ function, defined from the entropy function h , as

$$\kappa(\alpha) = 2[1 + h(\alpha)], \quad \text{with} \quad h(\alpha) = \alpha |\log \alpha| + (1 - \alpha) |\log(1 - \alpha)| \quad (1.2)$$

plays an important role in this work, notably in the analysis of **QuickQuant** $_{\alpha}$. Its graph is represented in Figure 1.5. The mean number $K(m, n)$ of key comparisons performed by **QuickSelect**(m, n), together with its variants is given in the following lemma. Its asymptotic estimate is presented in the column 4 of Table 1.6. The proof for the expression of $K(m, n)$ of **QuickSelect** was obtained by Knuth in [48].

Function QuickSelect($m, \text{left}, \text{right}$)	
/* Returns the key ranked m of the array $V[\text{left}..\text{right}]$	*/
if left < right then	
$k \leftarrow \text{Partition}(V, \text{left}, \text{right})$ /* Partition was defined in Quicksort	*/
if $k = m$ then return $V[k]$;	
if $k > m$ then QuickSelect($m, \text{left}, k - 1$)	
if $k < m$ then QuickSelect($m - k, n - k, \text{right}$)	
end	
<hr/>	
Function QuickMin(n)	
/* Returns the minimum of the array $V[1..n]$	*/
$k \leftarrow \text{Partition}(V, 1, n)$ /* Partition was defined in Quicksort	*/
if $k = 1$ then return $V[1]$ else return QuickMin($k - 1$)	

Figure 1.4: QuickSelect and QuickMin Algorithms. (N.B.: for clarity's sake, we denote by QuickSelect(m, n) a call to the procedure QuickSelect($m, 1, n$) described here.)

Lemma 1.5. *The mean number of key comparisons performed by QuickSelect(m, n) satisfies the following:*

- (a) *For the case of a general rank m , it involves the harmonic number H_k under the form*

$$K(m, n) = 2[(n+1)H_n - (n+3-m)H_{n+1-m} - (m+2)H_m + n+3], \quad (1.3)$$

In the case of QuickMin and QuickMax, one has respectively $K(1, n) = K(n, n) = 2n - 2H_n$.

- (b) *For the QuickQuant $_{\alpha}$ algorithm, the mean number $K_{\alpha}(n)$ involves the κ function defined in 1.2, under the form $K_{\alpha}(n) \sim \kappa(\alpha)n$. In particular,*
- (b1) *In the case of QuickMed, $K_{1/2}(n) \sim 2(1 + \log 2)n$.*
- (b2) *In the case of QuickRand(n), $K(n) \sim 3n$.*

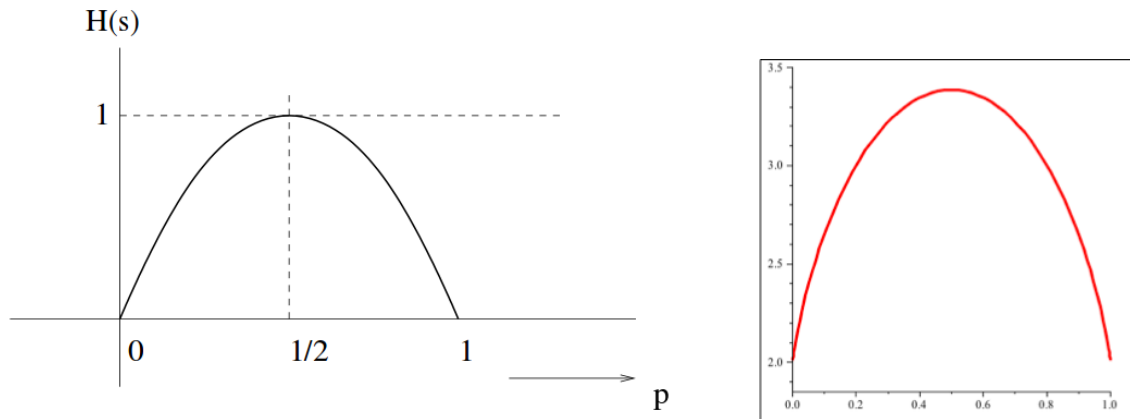


Figure 1.5: The functions h and κ

Proof. We first consider the particular cases $m = 1$ or $m = n$, and prove the second assertion of Part (a) of the Lemma. The case $m = 1$ corresponds to QuickMin algorithm.

Variants of <code>QuickSelect</code> (m, n)	Output	Value of m	Asymptotic estimate of $K(m, n)$
<code>QuickQuant</code> $_{\alpha}$	α -quantile	$m = \lfloor 1 + \alpha(n - 1) \rfloor$	$\kappa(\alpha)n$
<code>QuickMed</code>	median	$m = \lfloor \frac{n+1}{2} \rfloor$	$2(1 + \log 2)n$
<code>QuickMin</code>	minimum	$m = 1$	$2n$
<code>QuickMax</code>	maximum	$m = n$	$2n$
<code>QuickRand</code>	random	$m \in [1 \dots n]_{\mathcal{R}}$	$3n$

Figure 1.6: The asymptotic estimates for the mean number of key comparisons .

In this case, the number of key comparisons for the first partition is $n - 1$, and the recurrence is, for indices n and $n - 1$,

$$K(1, n) = n - 1 + \frac{1}{n} \sum_{i=2}^n K(1, i - 1), \quad K(1, n - 1) = n - 2 + \frac{1}{n - 1} \sum_{i=2}^{n-1} K(1, i - 1).$$

Then the difference $nK(1, n) - (n - 1)K(1, n - 1)$ is written as

$$nK(1, n) - (n - 1)K(1, n - 1) = n(n - 1) - (n - 1)(n - 2) + K(1, n - 1),$$

and leads to the recurrence relation

$$K(1, n) = K(1, n - 1) + 2 - \frac{2}{n}.$$

The solution is then $K(1, n) = 2n - 2H_n$, and, by symmetry, the equality $K(n, n) = 2n - 2H_n$ also holds. This proves the second assertion of Part (a) of the lemma.

For a general rank m , and in the case when the pivot has rank i , there are always two cases, according to the relative positions of the rank i of the pivot and the rank m of the key of interest. As previously, in the case of `QuickSort`, using conditional probabilities leads to the recurrence

$$K(m, n) = n - 1 + \frac{1}{n} (A(m, n) + B(m, n)).$$

which two involves auxilliary sequences

$$A(m, n) = \sum_{i=1}^{m-1} K(m - i, n - i) \quad \text{and} \quad B(m, n) = \sum_{i=m+1}^n K(m, i - 1).$$

These sequences satisfy the relations

$$A(m + 1, n + 1) - A(m, n) = K(m, n), \quad B(m, n + 1) - B(m, n) = K(m, n), \quad (1.4)$$

which allow us to eliminate the sequences A and B getting a “pure” recurrence for the sequence $K(m, n)$. We indeed remark the equality

$$\begin{aligned} (n + 1)K(m + 1, n + 1) - nK(m + 1, n) - nK(m, n) + (n - 1)K(m, n - 1) \\ = [(n - 1)n - 2n(n - 1) + (n - 1)(n - 2)] \\ + [A(m + 1, n + 1) - A(m + 1, n) - A(m, n) + A(m, n - 1)] \\ + [B(m + 1, n + 1) - B(m + 1, n) - B(m, n) + B(m, n - 1)] \\ = 2 + K(m + 1, n + 1) - K(m + 1, n) - K(m, n) + K(m, n - 1). \end{aligned}$$

(We used (1.4) to obtain the last equality.) We finally obtain a recurrence which only involves the sequence $K(m, n)$, namely

$$\begin{aligned} K(m+1, n+1) - K(m+1, n) - K(m, n) + K(m, n-1) \\ = [K(m+1, n+1) - K(m, n)] - [K(m+1, n) - K(m, n-1)] = \frac{2}{n+1}. \end{aligned}$$

The sequence $K(m+1, n+1) - K(m, n)$ satisfies the relation

$$\begin{aligned} K(m+1, n+1) - K(m, n) &= \frac{2}{n+1} + \frac{2}{n} + \cdots + \frac{2}{m+2} + K(m+1, m+1) - K(m, m) \\ &= 2(H_{n+1} - H_{m+1}) + K(m+1, m+1) - K(m, m). \end{aligned}$$

Iterating the relation, we obtain the general term $K(m, n)$ as a function of the particular sequences relative to $m = n$ or $m = 1$, namely

$$K(m, n) = 2 \sum_{k=2}^m (H_{n-m+k} - H_k) + K(1, n+1-m) + K(m, m). \quad (1.5)$$

Using the solution to these particular cases leads to the final solution

$$K(m, n) = 2[(n+1)H_n - (n+3-m)H_{n+1-m} - (m+2)H_m + n+3],$$

valid for $1 \leq m \leq n$. That proves Assertion (a) of the lemma.

We are interested in the asymptotic behaviour of the mean number $K_\alpha(n)$ of comparisons of the **QuickQuant** $_\alpha$ algorithm for $n \rightarrow \infty$. In this case, the index m equals $m = \lfloor \alpha(n-1) + 1 \rfloor = \alpha n + o(n)$. Applying Equation (1.3) and using the asymptotic behaviour of the harmonic number, namely $H_n = \log n + \gamma + o(1)$, we can neglect the terms of order $o(n)$ in

$$K(m, n) = 2[(n+1)H_n - (n+3-m)H_{n+1-m} - (m+2)H_m + n+3],$$

and write

$$\begin{aligned} K(m, n) &\sim 2[n(\log n + \gamma) - (1-\alpha)n(\log((1-\alpha)n) + \gamma) - \alpha n(\log(\alpha n) + \gamma) + n] \\ &\sim 2[n \log n + \gamma n - ((1-\alpha)n \log n + (1-\alpha)n \log(1-\alpha) + \gamma(1-\alpha)n)] \\ &\quad + 2[-\alpha n \log n - \alpha \log \alpha n - \alpha \gamma n + n] \\ &\sim 2[\gamma n - (1-\alpha)n \log(1-\alpha) + \gamma(1-\alpha)n - \alpha n \log \alpha - \alpha \gamma n + n] \\ &\sim 2n[(1-\alpha)|\log(1-\alpha)| + \alpha|\log \alpha| + 1]. \end{aligned}$$

With the entropy function $h(\alpha) = (1-\alpha)|\log(1-\alpha)| + \alpha|\log \alpha|$ and the expression of the κ function as a function of the entropy, as $\kappa(\alpha) = 2[1 + h(\alpha)]$, the mean number of key comparisons of **QuickQuant** $_\alpha$ has the asymptotic form:

$$K(m, n) = \kappa(\alpha)n.$$

Case of QuickRand. The **QuickRand** algorithm is the algorithm **QuickSelect** (m, n) where the rank m is chosen randomly in $[1..n]$. The asymptotic behaviour of the algorithm **QuickRand** can be obtained from the asymptotic behaviour of the algorithm **QuickQuant** (α) when α is uniformly chosen in the interval $[0, 1]$. Then, the mean number $K(n)$ of comparisons performed by **QuickRand** satisfies

$$K(n) \sim n \int_{\alpha=0}^1 \kappa(\alpha) d\alpha.$$

With the expression of κ given in (1.2) and the computation of the integral (by parts)

$$\int_0^1 x \log x dx = \left[\frac{x^2}{2} \log x \right]_0^1 - \frac{1}{2} \int_0^1 x dx = -\frac{1}{4},$$

the previous integral equals $2(1 + 1/4 + 1/4) = 3$. The mean number of key comparisons of QuickRand is then asymptotic to $3n$. \square

1.3.3 Mean number of local key-comparisons.

Lemma 1.6. *The mean number $\pi_n^{(m)}(i, j)$ of comparisons performed by the QuickSelect(m, n) algorithm between the keys of rank i and j (with $1 \leq i < j \leq n$) is*

$$\pi_n^{(m)}(i, j) = \frac{2}{\max(m, j) - \min(m, i) + 1}.$$

Proof. For the QuickMin algorithm, as for the QuickSort algorithm, the keys U_i and U_j are compared only if U_i or U_j is a pivot. This event coincides with the event “ U_i or U_j is the first key-in inside the subset $\mathcal{U}_{[1, j]}$ ” of probability $2/j$. Hence, the mean number of key comparisons $\pi_n(1, i, j)$ or $\pi_n(n, i, j)$ equals $2/j$.

For the QuickSelect(m, n) algorithm, we will prove the following fact: *Two keys U_i and U_j (with $i < j$) are compared if and only if one of them is chosen as the first pivot among keys in the set*

$$\mathcal{X} := \{U_\ell \in \mathcal{U}, \quad \ell \in [\min(i, m), \max(j, m)]\}.$$

The set \mathcal{X} is not modified until one of its elements U_ℓ is chosen as pivot. We assume that the chosen pivot U_ℓ is neither U_i nor U_j and we will prove that there is no comparison between U_i and U_j at this step and at the subsequent steps. There are three main cases, depending on the positions of m and ℓ with respect to the interval $[i \dots j]$.

First case $m \in [i \dots j]$ **and** $\ell \in]i \dots j[$. In this case, the interval $[\min(i, m) \dots \max(j, m)]$ coincides with the interval $[i \dots j]$. Due to the strategy of the algorithm, all the keys in \mathcal{Y} are compared to U_ℓ so that the array \mathcal{Y} is divided into two subarrays by U_ℓ , and U_i belong to the left subarray, whereas U_j belongs to the right subarray. There is no comparison between U_i and U_j at this step and at the subsequent steps.

Second case $m < i$ **and** $\ell \in]m, i[$. In this case, the interval $[\min(i, m) \dots \max(j, m)]$ coincides with the interval $[m \dots j]$. If $\ell \in]m \dots i[$, the array \mathcal{Y} is divided into two subarrays by U_ℓ . One contains U_m that is less than U_ℓ . The other one contains U_i and U_j that are both larger than U_ℓ . Due to the strategy of the algorithm which searches the key U_m , only the first subarray containing U_m is considered in the subsequent steps. Then, there is no comparison between U_i and U_j at this step and at the subsequent steps.

Third case $m > j$ **and** $\ell \in]j \dots m[$. In this case, the interval $[\min(i, m) \dots \max(j, m)]$ coincides with the interval $[i \dots m]$. If $\ell \in]j \dots m[$, \mathcal{Y} is subdivided into two subarrays by U_ℓ , one contains U_m and the other one contains U_i and U_j that are both less than U_ℓ . As the algorithm searches the key U_m , only the first subarray is considered in the next step. Then, there is no comparison between U_i and U_j at this step and at the subsequent steps. \square

The mean number of key comparisons in QuickMin is first easily recovered. One has

$$K(n) = 2 \sum_{j=1}^n \sum_{i=1}^{j-1} \frac{1}{j} = 2 \sum_{j=1}^n \frac{1}{j} (j-1) = 2n - H_n.$$

For the general **QuickSelect**(m, n) algorithm, we split the set of pairs (i, j) into three subsets, with respect to the point (m, m) , namely $j \leq m, i < m < j, i \geq m$

$$\begin{aligned} K(m, n) &= 2 \sum_{1 \leq i < j \leq n} \frac{1}{\max(m, j) - \min(m, i) + 1} \\ &= \sum_{1 \leq i < j \leq m} \frac{1}{m - i + 1} + \sum_{i < m < j} \frac{1}{j - i + 1} + \sum_{m \leq i < j \leq n} \frac{1}{j - m + 1}. \end{aligned}$$

The first sum is the cost of **QuickMax** for a size m , previously denoted by $K(m, m)$, whereas the third sum is the cost of **QuickMin** for a size $n + 1 - m$, denoted as $K(1, n + 1 - m)$. The middle sum is

$$\sum_{i=1}^{m-1} \sum_{j=m+1}^n \frac{2}{j - i + 1} = 2 \sum_{i=2}^m (H_{n-m+i} - H_i),$$

and we exactly recover the expression of $K(m, n)$ already obtained in (1.5). The computation is ended as above.

1.4 The InsertionSort algorithm.

1.4.1 Description

The Insertion Sort algorithm (**InsSort** in shorthand notation) is very natural. The algorithm uses the way people sort a hand of playing cards. One starts with an empty hand and the cards face down on the table. Then the cards are removed one by one from the table, and each removed card is inserted in a correct position inside the cards that are already on the hand. The insertion is done by comparing the new card *from right to left* with the cards already on the hand.

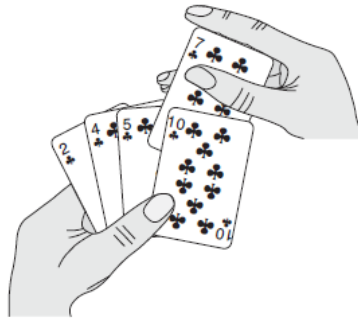


Figure 1.7: Sorting a hand of card using insertion sort. Figure from the source [17].

There are a sequence of $n - 1$ phases, indexed from $i = 2$ to $i = n$. At the i -th phase, there are two parts in the current array: the left subarray $W[1..i - 1]$ contains the keys of the initial subarray $V[1..i - 1]$ in the good order, whereas the right subarray is not yet modified and coincides with the initial subarray $V[i..n]$. During the i -th phase, the **InsSort** algorithm inserts the key V_i into the left sub-array $W[1..i - 1]$. As this subarray is already sorted, the right place for V_i is found by comparing V_i with the elements of the subarray $W[1..i - 1]$ *from right to left*.

```

Procedure InsSort( $V, n$ )
/* Sorts the array  $V[1..n]$  */
for  $i$  from 2 to  $n$  do
    for  $j$  from  $i$  downto 2 do
        if  $V[j-1] \geq V[j]$  then
            swap( $V[j], V[j-1]$ )
        end
        else Break /* exit the inner loop */
    end
end

```

Figure 1.8: InsSort Algorithm.

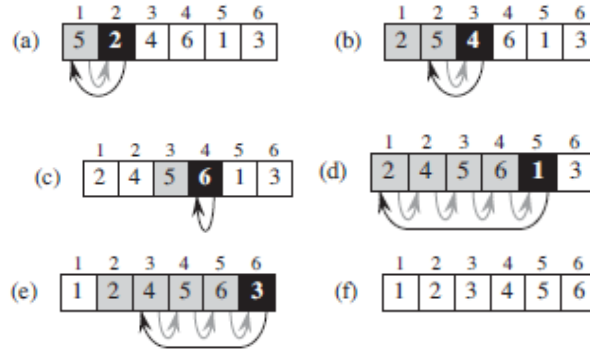


Figure 1.9: Illustration for InsSort

1.4.2 Mean number of key comparisons

Lemma 1.7. *The number of key comparisons performed by InsSort on an array of n keys is*

$$n-1 \quad (\text{best case}), \quad \frac{n(n-1)}{2} \quad (\text{worst-case}) \quad \frac{n(n-1)}{4} + n - H_n \quad (\text{average-case}).$$

Proof. The best-case arises when the array is already sorted. There are $n-1$ steps and the algorithm performs only one comparison at each step. The worst-case of InsSort arises when the initial array is sorted in the reverse order. There are exactly $i-1$ key comparisons during the i -th phase. Finally, the total number of comparisons is $\sum_{i=1}^{n-1} i = n(n-1)/2$.

Now we deal with the complexity on average case of InsSort, and we use the notion of inversion, that is recalled in 1.1.4. We remark that the swap of two adjacent items that are in the reverse order decreases the total number of inversions exactly by one. Then, the number of swaps performed by the algorithm equals the number of inversions. We have already analyzed the parameter “number of inversions” in Section 1.1.4, and we then deduce there are $n(n-1)/4$ such swaps on average. There are also the last comparison in each phase which is not followed with a swap, (except when the element is a left-to-right minimum). This is analyzed in the following result. There are $n - H_n$ such comparisons on average. \square

1.4.3 Mean number of local key comparisons.

Lemma 1.8. *The mean number $\pi_n(i, j)$ of comparisons performed by the **InsSort** algorithm between the keys of rank i and j (with $1 \leq i < j \leq n$) is*

$$\pi_n(i, j) = \frac{1}{2} + \frac{1}{(j-i+1)(j-i)}.$$

Proof. There are $n-1$ phases in the algorithm. During the i -th phase, the key V_i of the array is inserted into the left sub-array which contains an already sorted sequence built on the set $\{V_1, V_2, \dots, V_{i-1}\}$.

First case. U_i and U_j arrive in the wrong order in the initial array ($\tau(U_i) > \tau(U_j)$). In the phase when U_i is inserted into the left sub-array, this sub-array already contains U_j with $U_j > U_i$, and the key U_i is always compared and exchanged with U_j . This event is defined as “Inside the two keys set $\{U_i, U_j\}$, U_j is the first-in key, and U_i is the second-in key” and the probability of such an event is $1/2$ so that $\pi^-(i, j) = 1/2$.

Second case. U_i and U_j arrive in the good order in the initial array ($\tau(U_i) < \tau(U_j)$). The comparison does not always occur. In the phase when U_j is inserted into the left sub-array, this left sub-array already contains the key U_i . If this left sub-array contains one of the keys of the subset $\mathcal{U}_{[i,j]}$, then U_j “meets” (i.e., is compared to) this key before meeting U_i and remains on its right. Finally, the comparison between U_i and U_j occurs only if the subset $\mathcal{U}_{[i,j]}$ arrives after U_j . This defines the event “ U_i is the first-in key and U_j is the second-in key inside the set $\mathcal{U}_{[i,j]}$ ”. The probability of such an event and the expected value of symbol comparisons $\pi^+(i, j)$ is

$$\frac{1}{(j-i+1)(j-i)}.$$

□

The mean number of key comparisons performed by **InsSort** is then easily recovered. One has

$$\begin{aligned} K(n) &= \sum_{j=2}^n \sum_{i=1}^{j-1} \left[\frac{1}{2} + \frac{1}{(j-i)} - \frac{1}{(j-i+1)} \right] = \frac{n(n-1)}{4} + \sum_{j=2}^n \sum_{i=1}^{j-1} \left[\frac{1}{j-i} - \frac{1}{j-i+1} \right] \\ &= \frac{n(n-1)}{4} + \sum_{j=2}^n \left[1 - \frac{1}{j} \right] = \frac{n(n-1)}{4} + n - H_n \end{aligned}$$

1.5 The BubbleSort algorithm

1.5.1 Description.

Before 1962, **BubbleSort** (or briefly **BubSort**) had other names such as “sorting by exchange” [33] or “exchange sorting” [18, 30, 8]. It has been named “**BubbleSort**” by Iversen in 1962 in his book [43]. Even though Knuth writes that “**BubbleSort** seems to have nothing to recommend” [50], the algorithm is easy to remember, commonly used where the size n is not too large [67]. It is also easy to implement [63]. For these reasons, **BubSort** becomes a popular algorithm. We can learn more about the history of this algorithm in [3]. As its name says, the strategy of the algorithm is to push the smallest keys into the left of the array as the air bubbles on to the surface of a liquid. The algorithm performs $n-1$ phases. During one phase, the algorithm steps through the array, compares each pair

of adjacent keys and swaps them if they are in the wrong order. The i -th phase aims at finding the key of rank i and place it in the position i of the array. After the i -th phase, the keys of $\mathcal{U}_{[1..i]}$ are at their right places.

```

Procedure BubSort( $V, n$ )
/* Sorts the array  $V[1..n]$                                      */
for  $i$  from 1 to  $n - 1$  do
    for  $j$  from  $n$  downto  $i + 1$  do
        if  $V[j - 1] > V[j]$  then
            swap( $V[j - 1], V[j]$ )
        end
    end
end

```

Figure 1.10: BubSort algorithm.

Here is an example. The initial array is (23, 78, 45, 8, 32, 56). At each pass, keys

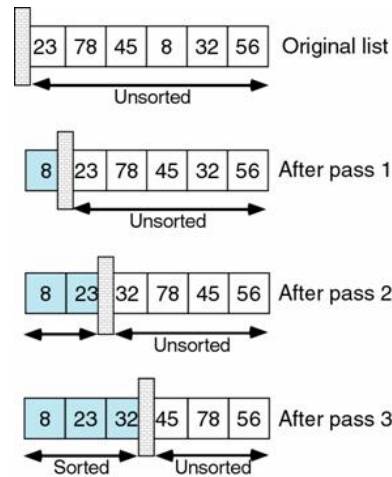


Figure 1.11: Illustration for BubSort. Figure from the source [75]

are swapped from the right to the left if they are not in the increasing order with their neighbor. After the first pass, the minimum of the array is found (8) and is positionned at the first place. After the second pass, the next relative minimum of the array is found and positionned at the second place, etc.

1.5.2 Mean number of comparisons.

Lemma 1.9. *The number of key comparisons performed by the BubSort algorithm on an array of n keys always equals $n(n - 1)/2$.*

This is clear: At the first step, there are $n - 1$ key comparisons. At the second phase, there are $n - 2$ key comparisons, and so on. In total, the number of key comparisons is $\frac{(n-1)n}{2}$.

1.5.3 Mean number of local key comparisons.

Lemma 1.10. *The mean number $\pi_n(i, j)$ of comparisons performed by the **BubSort** algorithm between the keys of rank i and j (with $1 \leq i < j \leq n$) is*

$$\pi_n(i, j) = \frac{1}{2} + \frac{1}{(j-i+1)(j-i)} + \frac{2(i-1)}{(j-i+2)(j-i+1)(j-i)}.$$

Proof. The **BubSort** algorithm may perform several comparisons between two keys U_i and U_j . We are first interested in the first comparison between U_i and U_j and we distinguish two cases:

First case. *U_i and U_j arrive in the right order in the initial array ($\tau(U_i) < \tau(U_j)$).* If there is one key of $\mathcal{U}_{i,j[}$ which arrives after U_i and before U_j , it will stay between U_i and U_j in the array thereafter, and will prevent U_i and U_j from meeting each other. If it arrives after U_j , it will eventually come between U_i and U_j in the array before these two keys meet each other. Hence, there is a comparison between U_i and U_j only if all the keys of the subset $\mathcal{U}_{i,j[}$ arrive before both U_i and U_j . This coincides with the event “the key U_j is the last-in and the key U_i arrived just before inside the subset $\mathcal{U}_{i,j[}$ ”. The probability that the first comparison between U_i and U_j occurs is

$$\frac{1}{(j-i+1)(j-i)}.$$

Second case. *U_i and U_j arrive in the wrong order in the initial array ($\tau(U_j) < \tau(U_i)$).* The first comparison between U_i and U_j occurs just before they are swapped. The probability of the event “ U_j is the first-in key and U_i is the second-in key in $\{U_i, U_j\}$ ” is $1/2$.

Subsequent comparisons. There might be subsequent comparisons between two keys. Note that, in both previous cases, immediately after the first comparison (either positive or negative) U_i and U_j are in the right order and in consecutive positions. A necessary condition for having at least one subsequent comparison between U_i and U_j is that all the keys of $\mathcal{U}_{i,j[}$ are still on the left of U_i after this point (for the same reasons exposed previously in Case 1). Now we also remark that any key U_ℓ with $\ell \in [1, i[$ which arrived after $\mathcal{U}_{i,j[}$ and before U_i in the first case, and after $\mathcal{U}_{i,j[}$ and before U_j in the second case, will be the cause of a stop of key U_i during some latter phases (such a key U_ℓ will never be swapped with U_i because of its smaller value). Also each time a key U_i is stopped during a phase by a key from $\mathcal{U}_{[1,i[}$, the set of keys from $\mathcal{U}_{[1,i[}$ between $\mathcal{U}_{i,j[}$ and U_i decreases by one during the same phase. After such a phase, as all keys to the right of U_i are in $\mathcal{U}_{[j,n]}$, the key U_j during the next phase will be swapped until reaching U_i (and results in a comparison). In conclusion the number of subsequent comparisons is exactly the number of keys from $\mathcal{U}_{[1,i[}$ which arrived after $\mathcal{U}_{i,j[}$ and before U_i in the first case and before U_j in the second case. For any $\ell \in [1 \dots i[$, the probabilities that U_ℓ arrives after $\mathcal{U}_{i,j[}$ and before U_i (and U_j arrives after U_i – Case 1) or after $\mathcal{U}_{i,j[}$ and before U_j (and U_i arrives after U_j – Case 2) have the same expression

$$\frac{1}{(j-i+2)(j-i+1)(j-i)}.$$

Using independence of events for $\ell \in [1, i[$, this yields that the mean number of subsequent (positive) comparisons (summing up for both Cases 1 and 2) is

$$\frac{2(i-1)}{(j-i+2)(j-i+1)(j-i)}.$$

To conclude, one has

$$\pi^+(i, j) = \frac{1}{(j-i+1)(j-i)} + \frac{2(i-1)}{(j-i+2)(j-i+1)(j-i)}, \quad \pi^-(i, j) = \frac{1}{2}.$$

□

It seems that the mean number of key comparisons performed by **InsSort** be not easily recovered in this case...

1.6 The Selection-Minimum Algorithm

The algorithm **SelMin** is the first phase of **SelectionSort**. This is the most natural strategy for finding the minimum key of an array. The variable called **Min** is initiated with the first key of the array. While stepping through the array, each key is compared with **Min** and replaces it if it is smaller. Then, the variable **Min** memorizes all the possible *déjà vu* minimum, namely the successive left to right minimum of the array. We recall that a left to right minimum of an array is the smallest key among all the keys which are on its left. It is clear that there are always $n - 1$ key comparisons.

Lemma 1.11. *The number of key comparisons performed by **SelMin** over an array of n keys is $n - 1$, in the best case, worst-case or average-case.*

Lemma 1.12. *The mean number of key comparisons between two keys U_i and U_j is*

$$\pi_n(i, j) = \frac{1}{i(i+1)} + \frac{1}{j(j-1)}.$$

Proof. If two keys U_i and U_j are compared, the first-in key of the set

$$\mathcal{U}_{[i,j]} = \{U_i, U_{i+1}, \dots, U_j\}$$

is a left to right minimum.

First case. U_i and U_j arrive in the right order ($\tau(U_i) < \tau(U_j)$). Then U_i is a left-to-right minimum, and U_j must arrive before the following left-to-right minimum, namely before all the keys of $\mathcal{U}_{[1,i]}$. Finally, inside the set $\mathcal{U}_{[1,i]} \cup \{U_j\}$, of cardinal $i + 1$, the key U_i is the first-in, and U_j is the second-in. Hence the number of key comparisons is $\pi^+(i, j) = 1/(i(i+1))$.

Second case. U_i and U_j arrive in the wrong order. ($\tau(U_i) > \tau(U_j)$). Then U_j is a left-to-right minimum and U_i is the following left-to-right minimum. This means that all the keys of the set $\mathcal{U}_{[1,i]} \cup \mathcal{U}_{[i,j]}$ arrived after U_i . Inside the set $\mathcal{U}_{[1,j]}$ of cardinal j , U_j is the first-in key, U_i is the second-in key. The mean number of key comparisons is $\pi^-(i, j) = 1/(j(j-1))$.

□

The mean number of key comparisons performed by **SelMin** is then easily recovered. One has

$$\begin{aligned} K(n) &= \sum_{i=1}^n \sum_{j=i+1}^n \frac{1}{i(i+1)} + \sum_{j=1}^n \sum_{i=1}^{j-1} \frac{1}{j(j-1)} = \sum_{i=1}^n (n-i) \left[\frac{1}{i} - \frac{1}{i+1} \right] + \sum_{j=1}^n \frac{1}{j} \\ &= \sum_{i=1}^n \frac{n-i}{i} - \sum_{i=2}^{n+1} \frac{n-i+1}{i} + H_n = (n-1) - \sum_{i=2}^n \frac{1}{i} + H_n = n-1 \end{aligned}$$

1.7 Towards a realistic analysis of selection and sorting algorithms.

1.7.1 Conclusion on the mean number of local comparisons.

We have seen in the previous sections the mean number of key comparisons performed by some sorting and selection algorithms. However, the purpose of this thesis is to find the mean number of symbol comparisons performed by these algorithms. For that, we first compute the mean number of local comparisons, denoted by $\pi_n(i, j)$. This is also the first step in the realistic analysis of selection and sorting algorithms that we will see in the general framework described in Chapter 5.

We summarize in the following proposition all the results that we have obtained in the chapter about the mean number $\pi_n(i, j)$ of (key)-comparisons between the key U_i of rank i and the key U_j of rank j .

Proposition 1.13. *Consider the five algorithms of interest. Then, in the permutation model, the mean number $\pi_n(i, j)$ of comparisons between the keys of rank i and j , with $i < j$ admits the expression described in the second column of Figure 1.12.*

Algorithms	$\pi_n(i, j)$	$K(n)$
QuickSort	$\frac{2}{j-i+1}$	$2n \log n$
InsSort	$\frac{1}{2} + \frac{1}{(j-i+1)(j-i)}$	$\frac{n^2}{4}$
BubSort	$\frac{1}{2} + \frac{1}{(j-i+1)(j-i)} + \frac{2(i-1)}{(j-i+2)(j-i+1)(j-i)}$	$\frac{n^2}{2}$
QuickSelect(m, n)	$\frac{2}{\max(m, j) - \min(m, i) + 1}$	$\Theta(n)$
QuickMin	$\frac{2}{j}$	$2n$
SelMin	$\frac{1}{i(i+1)} + \frac{1}{j(j-1)}$	n

Figure 1.12: Summary for the expressions of the mean number of local comparisons.

1.7.2 Normalized and non normalized algorithms.

Amongst these algorithms (and more generally amongst the sorting and searching algorithms), there are algorithms which perform a *systematic* comparison between keys that are in the *reverse* order. We say that these algorithms are *non-normalized*. Here, amongst the studied algorithms, there are only two *non-normalized* algorithms, namely, the **InsSort** and the **BubSort** Algorithms. Non-normalized algorithms may perform several comparisons between two keys in reverse order, but we only consider the first comparison of this type¹, and we define it as *the* non-normalized comparison. In summary, a comparison is *non-normalized* if this is a systematic comparison performed for the first time. Otherwise, it is said to be normalized.

¹This is important as **BubSort** may perform subsequent comparisons between these two keys.

For non-normalized algorithms, and in the permutation model, the non-normalized comparison arises with probability $1/2$, the probability that any two keys are in the reverse order. When the term $\pi_n^-(i, j)$ (associated with comparisons between keys in reverse order) contains a constant term $1/2$, this means that this algorithm is non-normalized, and we denote by $\hat{\pi}_n(i, j)$ the term $\pi_n(i, j)$ from which this constant is removed, namely

$$\hat{\pi}_n(i, j) := \begin{cases} \pi_n(i, j) - 1/2 & \text{for non-normalized algorithms} \\ \pi_n(i, j) & \text{for normalized algorithms.} \end{cases}$$

This notion will be important in Chapters 5 and 6: in our analysis, this term constant equal to $1/2$ plays a particular role and must be considered in a separate way.

Algorithms	$\hat{\pi}_n(i, j)$
QuickSort	$\frac{2}{j-i+1}$
InsSort	$\frac{1}{(j-i+1)(j-i)}$
BubSort	$\frac{1}{(j-i+1)(j-i)} + \frac{2(i-1)}{(j-i+2)(j-i+1)(j-i)}$
QuickSelect(m, n)	$\frac{2}{\max(m, j) - \min(m, i) + 1}$
QuickMin	$\frac{2}{j}$
SelMin	$\frac{1}{i(i+1)} + \frac{1}{j(j-1)}$

Figure 1.13: Summary for the expressions of the mean number of local normalized comparisons.

Towards a realistic analysis.

Replacing keys by words. In the thesis, when we perform the analysis of the main searching and sorting algorithms, we wish to take into account the possible internal structure of the inputs. In any computer, indeed, any input is a sequence of symbols (which belong to a given alphabet Σ), what is called a (finite) word, namely an element of Σ^* . When we compare two such words X and Y , it is natural to deal with the lexicographic order: the alphabet Σ is itself endowed with a given order, which is extended to Σ^* with the lexicographic order. The cost of such a comparison between two words X and Y is no longer a unit cost, and it now depends on the words themselves; it depends more precisely, on the coincidence between the two words. The coincidence $c(X, Y)$ between the two words X and Y is defined as the length of the longest common prefix of the two words. In this framework, the unit cost is now the comparison between symbols, and finally the cost for comparing two words X and Y equals $c(X, Y) + 1$, as the total number of symbol comparisons that are needed in the comparison of the two words.

From an intuitive point of view, the cost of comparison between two words (in terms of symbol comparisons) depends on their relative position, and in particular on their respective ranks. For instance, an algorithm which often compares keys U_i and U_j whose

ranks i and j are close will be probably less efficient (with respect to the number of symbol comparisons) than another algorithm which performs the same total number of key comparisons, but between keys whose ranks are less close. This is why the cost $\pi_n(i, j)$ (the mean number of comparisons between the key of rank i and the key of rank j) provides a first (but very useful) information in terms of the efficiency of the algorithm, when it will be further studied in terms of symbol comparisons.

Dealing with correlations between symbols. As keys are replaced by words which are formed with symbols, we will be interested in the mechanism which emits the symbols, namely a source. According to the source, successive emitted symbols may be strongly or weakly correlated. For example, symbols emitted by a simple source (memoryless source or Markov chain) are weakly correlated. They are independent in the case of memoryless source. The next Chapter will describe various models of source and introduces the general model we will choose for our analysis, the parameterized model.

Chapter 2

Sources

Contents

2.1	General sources	36
2.1.1	Generating functions of the source.	36
2.1.2	Entropy.	37
2.1.3	Coincidence.	37
2.2	Simple sources.	38
2.2.1	Memoryless sources.	38
2.2.2	Markov chains.	39
2.2.3	Other instances of “simple” sources: intermittent sources.	41
2.3	Dynamical sources	42
2.3.1	Definition of a dynamical source	42
2.3.2	Simple sources seen as dynamical sources	44
2.3.3	Transfer operators	46
2.4	Parametrization of a general source.	48
2.4.1	Non ambiguous sources.	49
2.4.2	Parametrization of a general source.	49
2.4.3	Geometry of the source	50

We have then decided to replace keys by words. The present chapter focuses on the probabilistic process that generates words, which is called in information theory a source. A source is a mechanism that emits symbols, which form words, that are given as inputs for the sorting or searching algorithms. The nature of the source may have a strong influence on the coincidence of words, and the coincidence between words is a central parameter in the analysis of this class of algorithms in terms of symbol comparisons.

The source will be itself the main input for our algorithms, and then plays a central role in the analysis of such algorithms. As it is usual in analysis of algorithms, we deal with generating functions, and the generating function of the source will be one of the main actors of our study. It involves fundamental probabilities, and is of Dirichlet type. And, asymptotic properties of the source will be related, as usual, to the position and the nature of these generating functions.

We first consider in this chapter various types of sources, simple sources (namely memoryless sources and Markov chains), but also more complex sources, where the correlation between symbols may depend on the whole previous history. We wish to describe a model

of sources, that would be sufficiently general, but we also wish to deal with sources whose Dirichlet generating function admits a nice expression, from which it would be possible to investigate the nature of the position of its singularities. The model of dynamical sources, introduced by Vallée and related to dynamical system theory goes beyond the cases of simple sources (memoryless sources and Markov chains). The transfer operator, a main tool in dynamical system, is here a main tool: it is used as a generating operator which itself generates... the generating function. These three classes of sources (memoryless sources, Markov chains and dynamical sources) will be our main instances, as their Dirichlet series are explicit.

However, our studies deal with a more general model of sources, which will be introduced in Section 2.1. It further extends the point of view provided by dynamical sources, and is based on a parametrization of the source. In this context, a source is just a random process defined on the unit interval, and the coincidence admits a natural representation, very useful in the sequel.

2.1 General sources

In information theory, a probabilistic source on the alphabet Σ is a mechanism which produces symbols from Σ , one at each time unit. When (discrete) time evolves in \mathbb{N} , a source produces (infinite) words in $\Sigma^{\mathbb{N}}$. More precisely, if X_k is the symbol emitted at time $t = k$, the source is defined by the sequence (X_k) of random variables with values in Σ . The complexity of the source is related to the correlations between these random variables (X_k) .

2.1.1 Generating functions of the source.

The emitted infinite word $(X_1, X_2, \dots, X_k, \dots)$ admits $X_1^k := (X_1, X_2, \dots, X_k)$ as its prefix of length k . The random variable X_1^k has its values which belong to the set Σ^k , and we are interested in the distribution of this random variable. The probability p_w that an infinite word begins with the finite prefix $w = w_1 w_2 \dots w_k$ is just

$$p_w := \Pr[X_1^k = w] = \Pr[X_1 = w_1, X_2 = w_2, \dots, X_i = w_i, \dots, X_k = w_k]. \quad (2.1)$$

The probability p_w is called the fundamental probability relative to the prefix w .

The set $\{p_w, w \in \Sigma^*\}$ defines completely the source \mathcal{S} , and it is natural to consider its generating function. As the set p_w is a set of probabilities, it has a multiplicative “flavour” and it proves useful to introduce Dirichlet series as generating functions.

Definition 2.1. *There are two main types of Dirichlet series associated to a source \mathcal{S} .*

(i) *The Λ generating functions are defined as*

$$\Lambda_k(s) = \sum_{w \in \Sigma^k} p_w^s, \quad \Lambda(s) = \sum_{k \geq 0} \Lambda_k(s) = \sum_{w \in \Sigma^*} p_w^s.$$

The bivariate generating function $\Lambda(s, u)$, where the variable u marks the length of the prefixes, is defined as

$$\Lambda(s, u) := \sum_{k \geq 0} u^k \Lambda_k(s) = \sum_{w \in \Sigma^*} u^{|w|} p_w^s.$$

(ii) The Π generating function is defined as

$$\Pi(s) = \sum_{k \geq 0} \pi_k^s, \quad \text{with} \quad \pi_k = \sup\{p_w, w \in \Sigma^k\}.$$

The Λ series was introduced for the first time in [71]. It plays a central role in many probabilistic studies about sources which adopt a point of view of analytic combinatorics. It also intervenes in this thesis, mainly in the analysis of sorting algorithms. The Π series was introduced for the first time in [12], and it is central here in the analysis of searching algorithms.

We remark the two equalities

$$\Lambda_k(1) = 1, \quad \forall k \geq 0; \quad \Pi_k(0) = 1, \quad \forall k \geq 0.$$

This proves that the series $\Lambda(s)$ is infinite at $s = 1$, and the series $\Pi(s)$ is infinite at $s = 0$. We will see that important probabilistic properties of the source can be expressed in terms of the regularity of Λ near $s = 1$ or the regularity of $\Pi(s)$ near $s = 0$.

2.1.2 Entropy.

Definition 2.2. The entropy $h(\mathcal{S})$ of the source \mathcal{S} is defined as the following limit (if it exists)

$$h(\mathcal{S}) = - \lim_{k \rightarrow \infty} \frac{1}{k} \mathbb{E}[\log p(X_1^k)], \quad (2.2)$$

where the random variable $p(X_1^k)$ is defined in Eq. (2.1).

Lemma 2.3. The entropy admits an alternative expression which involves the fundamental probabilities. It is also expressed with the Λ series

$$h(\mathcal{S}) = \lim_{k \rightarrow \infty} \frac{-1}{k} \sum_{w \in \Sigma^k} p_w \log p_w = \lim_{k \rightarrow \infty} \frac{-1}{k} \frac{d}{ds} \Lambda_k(s) |_{s=1}.$$

2.1.3 Coincidence.

In the previous chapter, a comparison between two keys has a unit cost. When the two keys are two (infinite) words built on an alphabet Σ , we are interested in a more realistic cost, which is equal to the number of symbols which are needed to distinguish these two (infinite) words. This cost is closely related to the coincidence, defined as follows.

Definition 2.4. The coincidence $c : \Sigma^{\mathbb{N}} \times \Sigma^{\mathbb{N}} \rightarrow \mathbb{N} \cup \{\infty\}$ is the random variable which measures the length of the longest common prefix between two infinite words X and Y ,

$$c(X, Y) := \max\{i; \quad X_j = Y_j, \quad \forall j \in [1..i]\}. \quad (2.3)$$

The number of symbols which are needed to distinguish the words X and Y is then equal to $c(X, Y) + 1$.

Lemma 2.5. The distribution of the coincidence between two independently drawn words admits an alternative expression which involves the fundamental probabilities. It is also expressed with the Λ series,

$$\Pr[c \geq k + 1] = \sum_{w \in \Sigma^k} p_w^2 = \Lambda_k(2), \quad \mathbb{E}[c] = \sum_{k \geq 1} \Pr[c \geq k] = \Lambda(2) - 1.$$

The mean number of symbols which is needed to distinguish two random words, called the coincidence of the source, and denoted by $c(\mathcal{S})$, is thus equal to $\Lambda(2)$.



Figure 2.1: Illustration of the coincidence between two words

Proof. The event $[c(X, Y) \geq k + 1]$ coincides with the event “The two words X and Y begin with the same prefix of length k ”, and thus with the disjoint union of the events, (taken over the words $w \in \Sigma^k$), “The two words X and Y begin with the prefix w ”. When the two words X and Y are independently drawn from the source, the probability of the last event equals p_w^2 . \square

2.2 Simple sources.

2.2.1 Memoryless sources.

Definition 2.6. A source \mathcal{S} is said to be memoryless if the variables X_k are independent with the same distribution. It is defined by the set p_i of probabilities, where p_i is the probability of emitting the symbol $i \in \Sigma$ at any time k , namely $p_i := \Pr[X_k = i]$ for any k .

Then, the generation of the k -th symbol X_k of the word does not depend on the previous emitted symbols. It is also independent of the time k when it is emitted.

The fundamental probability p_w that the word begins with the prefix $w = w_1 w_2 \dots w_k$ is then expressed as the product $p_w = \prod_{i=1}^k p_{w_i}$. This multiplicative property leads to an exact expression of the Dirichlet series

Lemma 2.7. In the memoryless case, the Λ Dirichlet series defined in (2.1), are expressed as a function of

$$\lambda(s) = \sum_{i \in \Sigma} p_i^s,$$

as

$$\Lambda_k(s) = \lambda(s)^k, \quad \Lambda(s) = \frac{1}{1 - \lambda(s)}, \quad \Lambda(s, u) = \frac{1}{1 - u\lambda(s)}. \quad (2.4)$$

Proof. We begin with the multiplicative property of p_w which extends to p_w^s . Then

$$\sum_{w \in \Sigma^k} p_w^s = \sum_{w \in \Sigma^k} \prod_{i=1}^k p_{w_i}^s = \prod_{i=1}^k \sum_{w_i \in \Sigma} p_{w_i}^s = \lambda(s)^k.$$

This proves the first relation in Equation (2.4). When summing over k the expression of $\lambda(s)^k$,

$$\Lambda(s) = \sum_k \sum_{w \in \Sigma^k} p_w^s = \sum_{k=0}^{\infty} \lambda(s)^k = \frac{1}{1 - \lambda(s)},$$

we get the second relation in Equation (2.4). \square

Lemma 2.8. In the memoryless case, the entropy of the source, defined in Equation (2.2), and the coincidence admit expressions that involve the function $\lambda(s)$, namely

$$h(\mathcal{S}) = -\lambda'(1) = -\sum_{i \in \Sigma} p_i \log p_i, \quad c(\mathcal{S}) = \frac{1}{1 - \lambda(2)} = \frac{1}{1 - \sum_i p_i^2}.$$

Proof. Taking the derivative with respect to s (at $s = 1$) of the expression provided in (2.4) provides,

$$\frac{d}{ds} \Lambda_k(s)|_{s=1} = k\lambda(1)^{k-1}\lambda'(1) = k\lambda'(1) = k \sum_i p_i \log p_i.$$

The definition of the entropy then entails the result. \square

In the memoryless case, and for an alphabet of a given cardinality r , the entropy is maximal when all the probabilities p_i are equal to $1/r$. In this case, the source is said to be unbiased. This maximal value equals $\log_2 r$. In the case of a binary alphabet $\Sigma = \{0, 1\}$, we let $p := p_0$ so that $p_1 = 1 - p$. The entropy is then simply denoted by $h(p)$ and is equal to

$$h(p) = -p \log p - (1 - p) \log(1 - p).$$

We already met this classical function in Chapter 1, and its graph is represented there, in Figure 1.5.

2.2.2 Markov chains.

A Markov chain source (of order 1) is a model of sources where the correlations between symbols may exist but are “weak”, as the emitted symbol can only be correlated with the previous symbol.

Definition 2.9. *A source on the alphabet Σ is a Markov chain of order 1 if and only if it satisfies the following : At each time k , and for each pair (i, j) of symbols of Σ^2 , the conditional probability of emitting i knowing that the previously emitted symbol is j does not depend on the time k , and is denoted by $p_{i|j}$*

$$\forall k \in \mathbb{N}, \quad \mathbb{P}[X_{k+1} = i \mid X_k = j] = p_{i|j}.$$

A Markov source is then completely defined by the vector V of initial probabilities $(v_i)_{i \in \Sigma}$ together with the transition matrix $\mathbf{P} := (p_{i|j})_{(i,j) \in \Sigma^2}$.

For example, with $\Sigma = \{0, 1\}$, and the initial probabilities $V = (p, 1 - p)$, the fundamental probability of the prefix $w = 001010$ is $p_w = p \cdot p_{0|0} \cdot p_{1|0}^2 \cdot p_{0|1}^2$.

Lemma 2.10. *Denote by \mathbf{P}_s the matrix with general coefficient $p_{i|j}^s$, and by V_s the vector of components v_i^s . Then the Dirichlet series of the Markov chain source are*

$$\Lambda_k(s) = {}^t \mathbf{1} \cdot \mathbf{P}_s^{k-1} \cdot V_s, \quad \Lambda(s) = 1 + {}^t \mathbf{1} \cdot (I - \mathbf{P}_s)^{-1} \cdot V_s, \quad \Lambda(s, u) = 1 + u {}^t \mathbf{1} \cdot (I - u \mathbf{P}_s)^{-1} \cdot V_s. \quad (2.5)$$

Proof. For $k \geq 1$, the coefficient of the matrix \mathbf{P}_s^{k-1} at the index (i, j) is the sum of all the terms of the form

$$p_{i_2|i_1}^s \cdot p_{i_3|i_2}^s \cdots p_{i_k|i_{k-1}}^s, \quad \text{with } i_1 = j, i_k = i, \text{ and } (i_2, \dots, i_{k-1}) \text{ any vector of } \Sigma^{k-2}.$$

Then, the matrix $\mathbf{P}_s^{k-1} \cdot V_s$ (that has r rows and one column) has in its i -th row the sum of all the terms of the form

$$v_j^s \cdot p_{i_2|j}^s \cdot p_{i_3|i_2}^s \cdots p_{i|i_{k-1}}^s, \quad \text{with } (j, i_2, \dots, i_{k-1}) \text{ any vector of } \Sigma^{k-1}.$$

This is the sum of all the terms of the form p_w^s where the prefix w of length k ends with the symbol i . To obtain the sum over all the prefixes of length k , we apply the matrix with

one row, and r columns, whose all its coefficients equal to 1, namely the matrix ${}^t\mathbf{1}$. We have then proven the equality

$$\Lambda_k(s) = {}^t\mathbf{1} \cdot \mathbf{P}_s^{k-1} \cdot V_s.$$

The first relation in Equation (2.5) is proven for all $k \geq 1$. We sum over k to get the second relation in Equation (2.5):

$$\Lambda(s) = 1 + \sum_{k \geq 1} \Lambda_k(s) = 1 + \sum_{k \geq 1} {}^t\mathbf{1} \cdot \mathbf{P}_s^{k-1} \cdot V_s = 1 + {}^t\mathbf{1} \cdot (I - \mathbf{P}_s)^{-1} \cdot V_s.$$

□

The coefficient (i, j) of the matrix \mathbf{P}^k , denoted by $P_{i,j}^k$, represents the probability of going from the state j to the state i in k time units. The matrix \mathbf{P} is irreducible if all the states communicate (possibly in an indirect way), i.e., if it is possible to go to any state from any state.

$$\forall i, \forall j, \quad \exists n, \quad \mathbf{P}_{i,j}^n > 0.$$

A state i is recurrent if there exists an integer $n = n(i, i) > 0$ such that $\mathbf{P}_{i,i}^n > 0$. A Markov chain is irreducible if all the states $i \in \Sigma$ are recurrent. The period of a recurrent state i is the length of the smallest cycle which contains i ; this is also equal to

$$d_i = \gcd\{n; \quad \mathbf{P}_{i,i}^n > 0\}.$$

The chain is aperiodic if the gcd $d := \gcd\{d_i, i \in \Sigma\}$ of all these periods is equal to 1.

In the thesis, the term “aperiodic” is used with another meaning. This is why we never use the term aperiodic with the previous meaning. A Markov chain whose transition matrix is irreducible and aperiodic will be said to be a good Markov chain. This will be said for the transition matrix too.

The Perron-Frobenius theorem [55] [34] states the following: *Consider a matrix \mathbf{T} with positive coefficients which is moreover good. Then, the matrix \mathbf{T} has a unique dominant eigenvalue λ , and a unique dominant eigenvector Π with positive components π_i whose sum equals 1.*

We apply this theorem to the matrix \mathbf{P}_s for any real s . Then, the matrix \mathbf{P}_s has a unique dominant eigenvalue $\lambda(s)$, and a unique dominant eigenvector Π_s with positive components $\pi_s^{(j)}$ whose sum equals 1. Since the matrix \mathbf{P} is stochastic, the dominant value $\lambda(s)$ satisfies $\lambda(1) = 1$, and the matrix $\mathbf{P} = \mathbf{P}_1$ has a unique (normalized) fixed vector $\Pi := \Pi_1$ with positive components $\pi^{(j)}$, whose sum equals 1.

Moreover, the matrix \mathbf{P}_s decomposes as a sum

$$\mathbf{P}_s = \lambda(s)\mathbf{Q}_s + \mathbf{R}_s,$$

where \mathbf{Q}_s is the projection on the dominant eigenspace, and \mathbf{R}_s is the remainder matrix, whose spectral radius $\rho(s)$ satisfies $\rho(s) := \max\{|\lambda|; \lambda \in \text{Sp}\mathbf{P}_s\} < |\lambda(s)|$. These matrices satisfy $\mathbf{Q}_s \cdot \mathbf{R}_s = \mathbf{R}_s \cdot \mathbf{Q}_s = 0$, so that the previous decomposition extends to any $k \geq 1$, namely

$$\mathbf{P}_s^k = \lambda^k(s)\mathbf{Q}_s + \mathbf{R}_s^k, \quad \text{and thus} \quad (I - \mathbf{P}_s)^{-1} = \frac{\lambda(s)}{1 - \lambda(s)}\mathbf{Q}_s + (I - \mathbf{R}_s)^{-1}.$$

This entails the following asymptotic behavior for the Λ Dirichlet series

$$\Lambda_k(s) = \lambda^{k-1}(s) [{}^t\mathbf{1} \cdot \mathbf{Q}_s \cdot V_s] + {}^t\mathbf{1} \cdot \mathbf{R}_s^k \cdot V_s = \lambda^k(s) w_s \left[1 + o(\rho^k) \right],$$

for some nonzero constant w_s and some $\rho < 1$. We remark that $w_1 = 1$. This also proves that $\Lambda(s)$ has a simple pole at $s = 1$. Moreover, there is a closed-form expression for the entropy.

Lemma 2.11. *In the case of a good Markov chain, with a transition matrix $\mathbf{P} = (p_{i|j})$, a dominant eigenvalue $\lambda(s)$, and a (normalized) fixed vector $\Pi := (\pi^{(j)})$ the entropy satisfies*

$$h(\mathcal{S}) = -\lambda'(1) = - \sum_{(i,j) \in \Sigma^2} \pi^{(j)} p_{i|j} \log p_{i|j}. \quad (2.6)$$

Proof. We first prove the equality $h(\mathcal{S}) = -\lambda'(1)$. This is obtained by taking the derivative of the estimate $\Lambda_k(s) = \lambda^k(s) w_s [1 + o(\rho^k)]$ with respect to k , namely

$$\frac{1}{k} \frac{d}{ds} \Lambda_k(s) \sim_{k \rightarrow \infty} \lambda'(s) \lambda^{k-1}(s) w_s \quad \text{and then} \quad \frac{1}{k} \frac{d}{ds} \Lambda_k(s)|_{s=1} \sim_{k \rightarrow \infty} \lambda'(1),$$

since the equality $w_1 = 1$ holds.

We then obtain an alternative expression for the derivative $\lambda'(1)$. Taking the derivative (with respect to s) of the equality $\mathbf{P}_s \cdot \Pi_s = \lambda(s) \Pi_s$ leads at $s = 1$ to the equality

$${}^t\mathbf{1} \cdot \mathbf{P}'_1 \cdot \Pi_1 + {}^t\mathbf{1} \cdot \mathbf{P}_1 \cdot \Pi'_1 = \lambda'(1) {}^t\mathbf{1} \cdot \Pi_1 + \lambda(1) {}^t\mathbf{1} \cdot \Pi'_1.$$

Moreover, since the matrix \mathbf{P} is stochastic, the equality ${}^t\mathbf{1} \cdot \mathbf{P}_1 = {}^t\mathbf{1}$ holds. This entails the expression for the entropy of the source given in (2.6). \square

2.2.3 Other instances of “simple” sources: intermittent sources.

Intermittent sources are an interesting particular case of a source of VLMC type (Variable Length Markov Chain), where the dependency from the past is unbounded. An intermittent source has two regimes, depending whether it emits a particular symbol $\underline{\sigma} \in \Sigma$ or not. Consider a source with an alphabet of finite cardinality $r \geq 2$. The source is *intermittent of exponent $a > 0$ with respect to $\underline{\sigma}$* if one has the following conditional probability distribution for the emission of each symbol in the word given the prefix preceding it. Define the event \mathcal{S}_k as $\mathcal{S}_k := \{\text{the prefix ends with a sequence of exactly } k \text{ occurrences of } \underline{\sigma}\}$. Then the conditional distribution of the next symbol emitted depends on the length k ; more precisely, one has $\Pr[\sigma | \mathcal{S}_0] = 1/r$ and, for $k \geq 1$,

$$\Pr[\underline{\sigma} | \mathcal{S}_k] = \left(1 - \frac{1}{k+1}\right)^a, \quad \Pr[\sigma | \mathcal{S}_k] = \left(1 - \left(1 - \frac{1}{k+1}\right)^a\right) \frac{1}{r-1} \quad \text{for } \sigma \neq \underline{\sigma}.$$

Then, in the case of a binary alphabet $\Sigma := \{0, 1\}$, when the source is intermittent with respect to 0, the probability of the prefixes 0^k and $0^k 1$ are respectively equal to

$$p_{0^k} = \frac{1}{2} \cdot \frac{1}{k^a}, \quad p_{0^k 1} = \frac{1}{2} \left(\frac{1}{k^a} - \frac{1}{(k+1)^a} \right),$$

and, with the language description $\{0, 1\}^* = (0^* 1)^* 0^*$, the series $\Lambda(s)$ admits the expression

$$\Lambda(s) = \frac{1 + 2^{-s} \zeta(as)}{1 - 2^{-s} [1 + \Sigma_a(s)]} \quad \text{with} \quad \Sigma_a(s) = \sum_{k \geq 1} \left[\frac{1}{k^a} - \frac{1}{(k+1)^a} \right]^s.$$

(Here, $\zeta(\cdot)$ is the Riemann zeta function.)

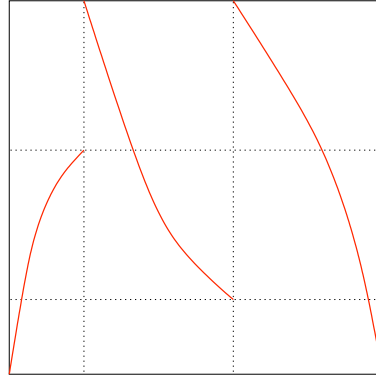


Figure 2.2: A Markovian dynamical system

2.3 Dynamical sources

A (probabilistic) dynamical source is defined by two objects: first, a symbolic mechanism, described by a dynamical system that associates an infinite word $M(x)$ to a real number x of the interval $[0, 1]$; second, a density on the unit interval.

2.3.1 Definition of a dynamical source

Definition 2.12. A dynamical system of the interval $\mathcal{I} := [0, 1]$ is defined by a mapping $T : \mathcal{I} \rightarrow \mathcal{I}$ (called the shift), for which

- (a) There exists a finite alphabet Σ and a family of disjoint open intervals \mathcal{I}_m indexed by Σ which define a topological partition of \mathcal{I} ; namely,

$$\exists (\mathcal{I}_m)_{m \in \Sigma}, \quad \mathcal{I} = \bigcup_{m \in \Sigma} \mathcal{I}_m.$$

- (b) The restriction $T_{[m]}$ of T to each interval \mathcal{I}_m is a \mathcal{C}^2 injection from \mathcal{I}_m to \mathcal{I} .

The system is complete when each restriction $T_{[m]}$ is surjective, .i.e, $T(\mathcal{I}_m) = \mathcal{I}$. The system is Markovian when each interval $T(\mathcal{I}_m)$ is a union of intervals \mathcal{I}_j .

In the sequel, we restrict ourselves to the case of a complete mapping, With this dynamical system, we build a probabilistic source as follows.

Dynamical source. The mapping $\tau : \mathcal{I} \rightarrow \Sigma$ which is equal to symbol m on each open interval \mathcal{I}_m will be useful for encoding the trajectories. To each $x \in \mathcal{I}$, one first associates the trajectory $(x, T(x), T^2(x), \dots, T^j(x)) \in \mathcal{I}^{\mathbb{N}}$, then the (infinite) word $M(x) \in \Sigma^{\mathbb{N}}$ formed with the successive symbols:

$$M(x) = (m_1(x), m_2(x), \dots, m_n(x), \dots), \quad \text{with} \quad m_j(x) = \tau(T^{j-1}(x)).$$

When the unit interval \mathcal{I} is endowed with a density f , and its associate distribution function F , the mapping M becomes a random variable $M : \mathcal{I} \rightarrow \Sigma^{\mathbb{N}}$. A dynamical system, together with a density f on the unit interval \mathcal{I} defines a *probabilistic dynamical source*.

For $m \in \Sigma$, the interval $\mathcal{I}_m = \tau^{-1}(\{m\})$ is the subset of $x \in \mathcal{I}$ for which the word $M(x)$ begins with m . If we denote by $h_{[m]} : \mathcal{I} \rightarrow \mathcal{I}_m$ the inverse of $T_{[m]}$, this interval \mathcal{I}_m

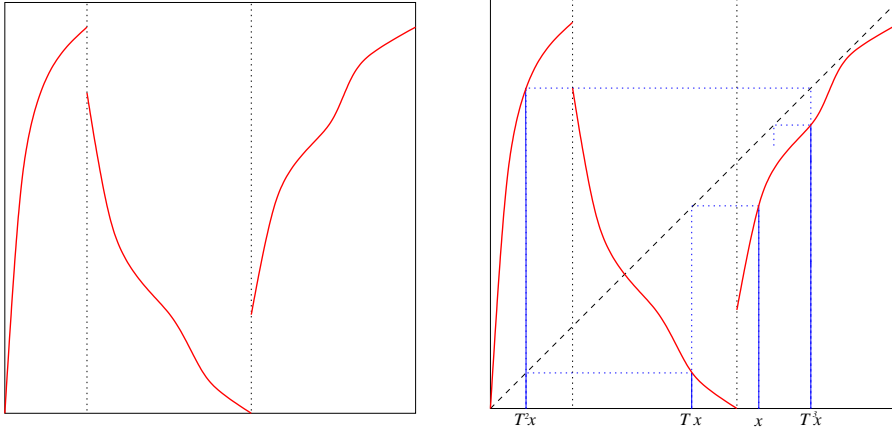


Figure 2.3: A dynamical system, with $\Sigma = \{a, b, c\}$ and a word $M(x) = (c, b, a, c, \dots)$

is also equal to $h_{[m]}(\mathcal{I})$. For any integer $k \geq 1$, each local inverse of the k -iterate T^k is said to be of depth k ; it is associated to a word $w = m_1 m_2 \dots m_k \in \Sigma^k$, and is of the form $h_{[w]} = h_{[m_1]} \circ h_{[m_2]} \circ \dots \circ h_{[m_k]}$. The set of all the inverse branches of depth k is then

$$\mathcal{H}^k = \{h_{[w]}, w \in \Sigma^k\}, \quad \text{and we let} \quad \mathcal{H}^* := \bigcup_{k \geq 0} \mathcal{H}^k = \{h_{[w]}, w \in \Sigma^*\},$$

so that \mathcal{H}^* is the set of all inverse branches of any depth.

Fundamental intervals. Consider more generally, for any $w \in \Sigma^*$, the interval \mathcal{I}_w , defined as the transform of the unit interval \mathcal{I} by the inverse branch $h_{[w]}$, namely $\mathcal{I}_w = h_{[w]}(\mathcal{I})$. By definition of the word $M(x)$, this interval \mathcal{I}_w gathers all the $x \in \mathcal{I}$ for which the word $M(x)$ begins with the prefix w . Then, the fundamental probability p_w (defined in (2.1) is expressed with the inverse branch $h_{[w]}$ and the distribution F of the density f , namely

$$p_w = \int_{\mathcal{I}_w} f(t) dt = \int_{h_{[w]}(\mathcal{I})} f(t) dt = |F(h_{[w]}(1)) - F(h_{[w]}(0))|. \quad (2.7)$$

Possible correlations. A dynamical source may have a high degree of correlations, due to both the geometry and the shape of its branches.

The geometry of the branches is defined by the position of horizontal intervals with respect to vertical intervals $\mathcal{J}_l = T(\mathcal{I}_l)$. This provides a first access to the correlation between successive symbols and allows to describe the set S_m formed with symbols that can be possibly emitted after symbol m . In a complete system, any symbol of Σ can be emitted after any symbol m so that $S_m = \Sigma$. In Markovian system, the set of symbols that can be possibly emitted after the symbol m is denoted by K_m , and thus $S_m = K_m$.

The shape of the branches, more precisely, the behavior of the derivatives $h'_{[m]}$ has also a great influence on correlations between symbols. For a fixed geometry of branches, a system with affine branches is less correlated than other systems with the same geometry. We will see that simple sources are defined by systems with affine branches, and uniform initial (local) density.

When the branches of the shift T are all increasing or all decreasing, the shift T is said to be “homoclinic”. Otherwise, the shift T is “heteroclinic”. As we will see, the

dynamical systems of continued fraction source and binary numeration are homoclinic, and the dynamical system underlying simple sources may be heteroclinic or homoclinic.

2.3.2 Simple sources seen as dynamical sources

Simple sources (memoryless, or Markov chains sources) can be viewed as dynamical source: They are associated to dynamical systems, with affine branches and uniform initial distribution. When the derivatives h' of the branches are not constant, this creates correlations between successive symbols and the sources are no longer simple.

Memoryless sources. A memoryless source is a dynamical source which satisfies the two properties:

- (a) It is associated to a complete dynamical system whose each restriction $T|_{\mathcal{I}_m}$ of T to \mathcal{I}_m is affine (increasing or decreasing)
- (b) Its initial density is uniform.

Figure 2.4 is an example of a memoryless sources associated to the three intervals of respective lengths $p_1 = 1/6, p_2 = 1/3, p_3 = 1/2$ and equal to

$$\mathcal{I}_1 = [0, 1/6], \quad \mathcal{I}_2 = [1/6, 1/2], \quad \mathcal{I}_3 = [1/2, 1].$$

The shift T is defined by

$$T|_{\mathcal{I}_1} : x \mapsto 6x, \quad T|_{\mathcal{I}_2} : x \mapsto 3x - \frac{1}{2}, \quad T|_{\mathcal{I}_3} : x \mapsto 2x - 1.$$

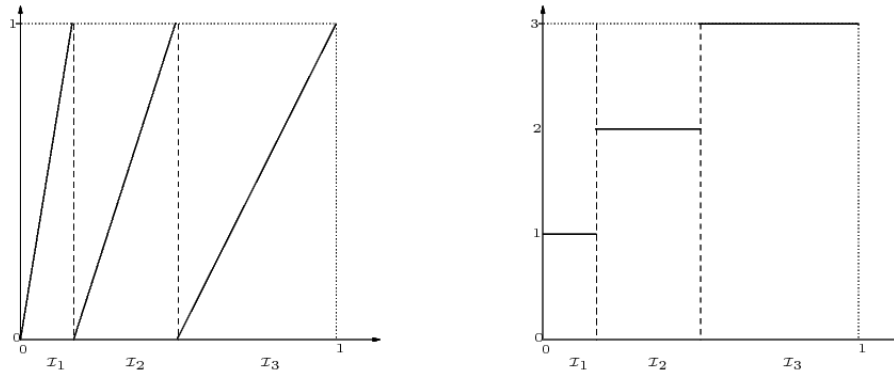


Figure 2.4: Memoryless sources: On the left the shift T . On the right, the encoding τ .

Numeration system in base b . The shift of the numeration system of base b is

$$T(x) = \{bx\} = bx - \lfloor bx \rfloor, \quad \tau(x) = \lfloor bx \rfloor.$$

This transformation computes the expansion in base b of the real x

$$x = \sum_{j=1}^{\infty} m_j b^{-j}, \quad \text{with } m_j := \tau(T^j(x)).$$

For example, in the case $b = 2$, the shift T is defined as

$$T(x) = 2x, \quad x \in \left[0, \frac{1}{2}\right] \quad \text{and} \quad T(x) = 1 - 2x, \quad x \in \left[\frac{1}{2}, 1\right],$$

and, each end of a fundamental interval admits two expansions; for instance, for $x = \frac{1}{4}$, there are two distinct words $M(x)$,

$$N\left(\frac{1}{4}\right) = 0.01000\dots \quad \text{or} \quad N\left(\frac{1}{4}\right) = 0.001111\dots$$

The numeration system of base 2 is illustrated by Figure 2.5. When the initial density is uniform, this gives rise to an unbiased memoryless source.

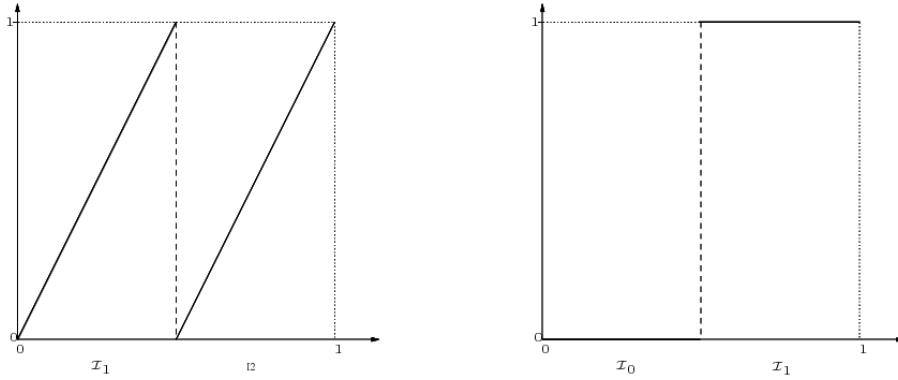


Figure 2.5: Numeration system of base 2

Markov chains. A Markov chain is a dynamical source which satisfies the two properties:

- (a) It is associated to a Markovian dynamical system whose each restriction $T_{[m]}$ of T to \mathcal{I}_m is affine (increasing or decreasing)
- (b) Its initial density is constant on each \mathcal{I}_m .

Figure 2.6 illustrates how a Markov chain may be viewed as a dynamical source.

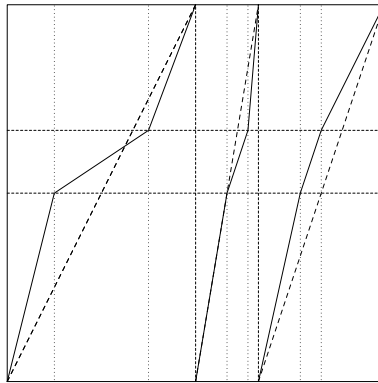


Figure 2.6: a Markov chain, viewed as dynamical sources.

The continued fraction source. The continued fraction source is an instance of a dynamical source for which correlations between symbols may depend on the whole past. Here, as its alphabet is the set of integers \mathbb{N} , it does not enter the framework of this

thesis, where the alphabet Σ is always finite. The topological partition of \mathcal{I} is defined by $\mathcal{I}_m =]1/(m+1), 1/m[$, i and the shift is defined as

$$T(x) = \left\{ \frac{1}{x} \right\} = \frac{1}{x} - \left\lfloor \frac{1}{x} \right\rfloor, \quad \tau(x) = \left\lfloor \frac{1}{x} \right\rfloor.$$

The restriction of the shift T to the interval \mathcal{I}_m is $T(x) = (1/x) - m$, and the inverse branch $h_{[m]}$ is defined on the interval \mathcal{I} by $h_{[m]}(x) = 1/(x+m)$. The iteration of this transformation gives rise to the continued fraction expansion of x .

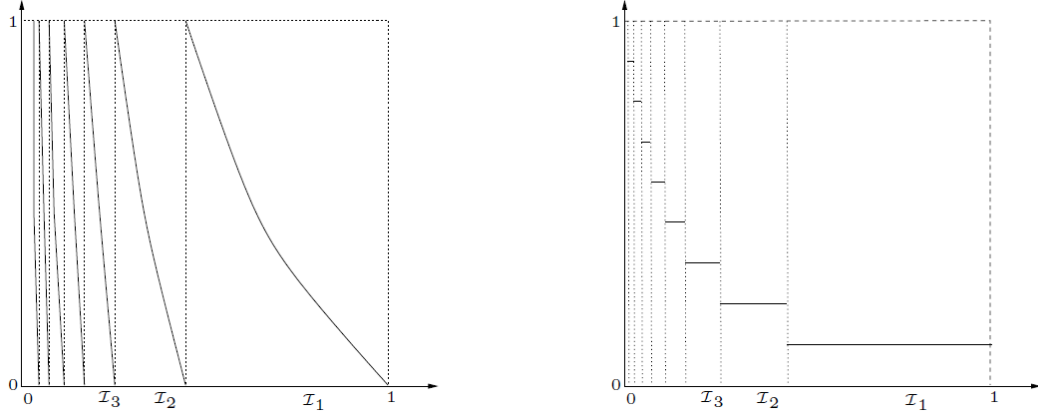


Figure 2.7: The Continued fraction source: on the left the shift T , on the right, the encoding application τ .

2.3.3 Transfer operators

One of the main tool in dynamical system theory is the transfer operator, introduced by Ruelle [58], denoted by \mathbf{H}_s . It generalizes the density transformer \mathbf{H} that describes the evolution of the density. A generalized version of the transfer operator is the secant operator \mathbb{H}_s that gives rise to an expression of the Dirichlet series $\Lambda(s)$ as a quasi inverse, in a way that generalizes expressions obtained in Equation (2.4), (2.5).

Density transformer. The density transformer associated to the dynamical system defined by (\mathcal{I}, T) describes the evolution of densities on \mathcal{I} under iteration of T : if X is a random variable with density f , what is the density of the iterate TX ? We define the density transformer \mathbf{H} as

$$\mathbf{H}[f](x) := \sum_{i \in \Sigma} |h'_{[i]}(x)| \cdot f \circ h_{[i]}(x) = \sum_{h \in \mathcal{H}} |h'(x)| \cdot f \circ h(x).$$

This is an operator which transforms a function defined on the interval \mathcal{I} into another function $\mathbf{H}[f]$, also defined on the interval \mathcal{I} . In the case of a complete dynamical system, if f_0 is the initial density on \mathcal{I} and f_1 is the density on \mathcal{I} after one iteration of T , then f_1 can be written as $f_1 = \mathbf{H}[f_0]$, where \mathbf{H} is defined as

$$\mathbf{H} = \sum_{h \in \mathcal{H}} \mathbf{H}_{(h)}, \quad \text{with,} \quad \mathbf{H}_{(h)}[f](x) = |h'(x)| f \circ h(x).$$

Transfer operators. The transfer operator extends the density transformer. One adds a complex parameter s , and replace the derivatives $|h'(x)|$ by $|h'(x)|^s$; we consider

$$\mathbf{H}_s[f] := \sum_{h \in \mathcal{H}} |h'(x)|^s \cdot f \circ h(x) = \sum_{i \in \Sigma} |h'_{[i]}(x)|^s \cdot f \circ h_{[i]}(x), \quad (2.8)$$

where, as previously, the set \mathcal{H} is the set of all inverse branches h of T . Multiplicative properties of derivatives prove that the k -th iterate of the transfer operator involves the set \mathcal{H}^k under the form:

$$\mathbf{H}_s^k[f](x) = \sum_{h \in \mathcal{H}^k} |h'(x)|^s \cdot f \circ h(x) = \sum_{w \in \Sigma^k} |h'_{[w]}(x)|^s \cdot f \circ h_{[w]}(x).$$

For obtaining an alternative expression of fundamental probabilities defined in Equation (2.7), it proves convenient to introduce the secant operator which is a further generalization of the Ruelle operator introduced by Vallée [70]. This operator replaces tangents of inverse branches by their secants, and considers

$$\left| \frac{h(x) - h(y)}{x - y} \right|^s \quad \text{instead of} \quad |h'(x)|^s.$$

Such an operator acts on functions F of two variables, and is defined as

$$\mathbb{H}_s[F](x, y) := \sum_{h \in \mathcal{H}} \left| \frac{h(x) - h(y)}{x - y} \right|^s \cdot F(h(x), h(y)). \quad (2.9)$$

The secant operator is an extension of the plain transfer operator, since, on the diagonal $x = y$, one has

$$\mathbb{H}_s[F](x, x) = \mathbf{H}_s[\text{diag } F](x), \quad (2.10)$$

where the function $\text{diag } F$ is defined by $\text{diag } F(x) := F(x, x)$.

Multiplicative properties of secants prove that the k -th iterate of \mathbb{H} admits the expression

$$\mathbb{H}_s^k[F](x, y) = \sum_{h \in \mathcal{H}^k} \left| \frac{h(x) - h(y)}{x - y} \right|^s F(h(x), h(y)) = \sum_{w \in \Sigma^k} \left| \frac{h_{[w]}(x) - h_{[w]}(y)}{x - y} \right|^s F(h_{[w]}(x), h_{[w]}(y)). \quad (2.11)$$

Expression of the Λ series. Moreover, for $w \in \Sigma^k$, the probability p_w^s can be written as

$$p_w^s = |F(h_{[w]}(1)) - F(h_{[w]}(0))|^s = \left| \frac{F(h_{[w]}(1)) - F(h_{[w]}(0))}{h_{[w]}(1) - h_{[w]}(0)} \right|^s \cdot \left| \frac{h_{[w]}(1) - h_{[w]}(0)}{1 - 0} \right|^s.$$

This is a product of two factors which each involves a secant, the secant of the inverse branch $h_{[w]}$, and the secant L of the distribution F related to the initial density f , namely

$$p_w^s = \left| \frac{h_{[w]}(1) - h_{[w]}(0)}{1 - 0} \right|^s \cdot L^s(h_{[w]}(0), h_{[w]}(1)), \quad \text{with} \quad L(x, y) := \left| \frac{F(x) - F(y)}{x - y} \right|. \quad (2.12)$$

The comparison of the two relations given in (2.11) and in (2.12) is a main step for the following result which provides an alternative expression of the Λ series as a function of the secant transfer operator:

Lemma 2.13. *Consider a complete dynamical system defined from a dynamical system (\mathcal{I}, T) together with an initial density f . Then, the Λ Dirichlet series are expressed with the secant L of the distribution F related to the density f , together with the secant transfer operator \mathbb{H}_s defined as (2.9), namely*

$$\Lambda_k(s) := \sum_{w \in \Sigma^k} p_w^s = \mathbb{H}_s^k[L^s](1, 0), \quad \Lambda(s) = (1 - \mathbb{H}_s)^{-1}[L^s](1, 0), \quad (2.13)$$

$$\Lambda(s, u) = (1 - u\mathbb{H}_s)^{-1}[L^s](1, 0).$$

Proof. We have already proven the expression for $\Lambda_k(s)$ as a function of the iterate \mathbb{H}_s^k by comparing (2.11) and (2.12). The sum over k leads to the quasi-inverse of \mathbb{H}_s as it is stated in (2.13). \square

Transfer operator of memoryless sources. In the case of the memoryless sources, the function $\mathbf{H}_s[\mathbf{1}](x)$ is a constant function, and for any $x \in \mathcal{I}$, one has

$$\mathbf{H}_s[\mathbf{1}](x) = \sum_{h \in \mathcal{H}} |h'(x)|^s = \sum_{m \in \Sigma} |h'_{[m]}(x)| = \sum_{m \in \Sigma} p_m^s = \lambda(s).$$

Then, $\lambda(s)$ is the eigenvalue of \mathbf{H}_s for the eigenfunction $\mathbf{1}$ and $\Lambda(s) = 1/(1 - \lambda(s))$ is the eigenvalue of $(I - \mathbf{H}_s)^{-1}$ for the eigenfunction $\mathbf{1}$. In the same vein, $\lambda(s)$ is the eigenvalue of \mathbb{H}_s , relative to the constant function $\underline{\mathbf{1}}$, equal to 1 on the square \mathcal{I}^2 and $\Lambda(s) = 1/(1 - \lambda(s))$ is the eigenvalue of $(I - \mathbf{H}_s)^{-1}$ for the same function $\underline{\mathbf{1}}$.

Good Class. Here we consider particular complete dynamical systems, for which it is possible to prove that the quasi-inverse has nice spectral properties on a convenient functional space. This will entail nice properties for the $\Lambda(s)$ series.

Definition 2.14. *A dynamical system of the interval (\mathcal{I}, T) belongs to the Good Class if it is complete, with a set \mathcal{H} of inverse branches which is uniformly contracting, i.e, there exists a constant $\rho < 1$ (called the contraction ratio) for which*

$$\forall h \in \mathcal{H}, \quad \forall x \in \mathcal{I}, \quad |h'(x)| \leq \rho.$$

When the dynamical system belongs to the Good Class, the transfer operator (acting on a convenient functional space) behaves as a finite matrix, and all what we have said for the transition matrix \mathbf{P}_s of a good Markov chain can be extended to the transfer operator \mathbb{H}_s : It admits dominant spectral properties for s near the real axis, together with a spectral gap. This implies the following:

Lemma 2.15. *In the case of a dynamical source of the Good Class, the function $\Lambda(s)$ is analytic on the half plane $\{\Re s > 1\}$ and admits a simple pole at $s = 1$ with a residue equal to $-1/\lambda'(1)$. Moreover the entropy of the source is well defined and expressed with the dominant eigenvalue $\lambda(s)$ of the transfer operator \mathbb{H}_s as $h(\mathcal{S}) = -\lambda'(1)$.*

2.4 Parametrization of a general source.

In the case of a dynamical system, we have associated to a real $x \in \mathcal{I}$ a word $M(x)$ that is the encoding of the trajectory $\mathcal{T}(x)$. We have also built a family of fundamental intervals \mathcal{I}_w , whose measures are the fundamental probabilities of the source. Moreover, for each depth k , the fundamental intervals \mathcal{I}_w form a topological partition of the interval \mathcal{I} .

A general source is completely defined by the family (p_w) . We will now adopt the inverse point of view and wish to associate to this source a family of fundamental intervals \mathcal{I}_w whose probability is equal to p_w and a parametrization $M : \mathcal{I} \rightarrow \Sigma^{\mathbb{N}}$, which will extend the point of view of dynamical sources.

2.4.1 Non ambiguous sources.

Here, and in all the sequel, we restrict ourselves to a class of particular sources, that gathers *non ambiguous* sources.

Definition 2.16. *Let Σ be a totally ordered alphabet of cardinality r . A source over the alphabet Σ produces infinite words of $\Sigma^{\mathbb{N}}$, and is specified by the fundamental probabilities p_w , $w \in \Sigma^*$, where p_w is the probability that an infinite word begins with the finite prefix w . When the two following properties hold,*

(i) $p_w > 0$ for any $w \in \Sigma^*$, (ii) $\pi_k := \max\{p_w : w \in \Sigma^k\}$ tends to 0, as $k \rightarrow \infty$, the source is said to be non-ambiguous.

In the sequel, all the sources are assumed to be non-ambiguous.

2.4.2 Parametrization of a general source.

We will now build the fundamental intervals and the parametrization M which will extend their “dynamical” analogs. These objects will depend on the order which is defined on the alphabet.

The sets Σ^k , for $k \geq 1$ and the set $\Sigma^{\mathbb{N}}$ are endowed with the strict lexicographic order (associated to the order on Σ) and denoted by ‘ $<$ ’. For any prefix $w \in \Sigma^*$, we denote by $|w|$ the length of w (i.e., the number of the symbols that it contains) and a_w, b_w, p_w the probabilities that a word produced by the source begins with a prefix α of the same length as w , which satisfies $\alpha < w$, $\alpha \leq w$, or $\alpha = w$, meaning

$$a_w := \sum_{\substack{\alpha, |\alpha|=|w|, \\ \alpha < w}} p_\alpha, \quad b_w := \sum_{\substack{\alpha, |\alpha|=|w|, \\ \alpha \leq w}} p_\alpha, \quad p_w = b_w - a_w. \quad (2.14)$$

Thus, for a given k , when the prefix w varies in Σ^k , this gives rise to a partition of the unit interval with subintervals of length p_w . When the prefixes $w \in \Sigma^k$ are ordered in increasing lexicographic order, and the subintervals are arranged from left to right, then, the subinterval corresponding to prefix w has left (respectively, right) endpoint equal to a_w (resp., b_w). See Figure 2.8.

Consider the set $\Sigma^{\mathbb{N}}$ of (infinite) words produced by the source \mathcal{S} , ordered via the lexicographic order. Given an infinite word $X \in \Sigma^{\mathbb{N}}$, denote by w_k its prefix of length k . The sequence $(a_{w_k})_{k \geq 0}$ is increasing, the sequence $(b_{w_k})_{k \geq 0}$ is decreasing, and $b_{w_k} - a_{w_k} = p_{w_k}$ tends to 0 for $k \rightarrow \infty$. Thus a unique real $P(X) \in [0, 1]$ is defined as the common limit of (a_{w_k}) and (b_{w_k}) , and $P(X)$ is simply the probability that an infinite word Y be smaller than X . The mapping $P : \Sigma^{\mathbb{N}} \rightarrow [0, 1]$ is surjective and strictly increasing outside the exceptional set formed with words of $\Sigma^{\mathbb{N}}$ which end with an infinite sequence of the smallest symbol or with an infinite sequence of the largest symbol.

Conversely, almost everywhere (except on the set $\{a_w, w \in \Sigma^*\}$), there is a mapping M which associates, to a number u of the interval $\mathcal{I} := [0, 1]$, a word $M(u) \in \Sigma^{\mathbb{N}}$. Hence, the probability that a word Y be smaller than $M(u)$ equals u . The lexicographic order on words

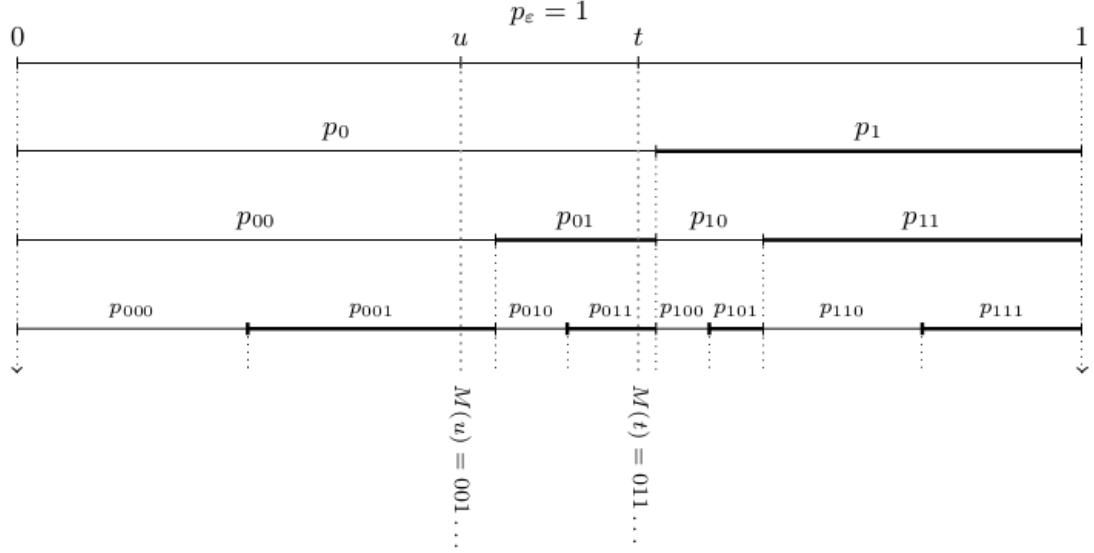


Figure 2.8: The parametrization of a source

(‘<’) is then compatible with the natural order on the interval \mathcal{I} , namely, $M(t) \leq M(u)$ if and only if $t \leq u$.

In Figure 2.8, we see the representation of two words of respective parameters u and t . For each depth ℓ , the prefix of length ℓ of a word $M(u)$ is by definition the prefix w for which u belongs to the fundamental interval $[a_w, b_w]$.

The symmetric source. The symmetric source $\check{\mathcal{S}}$ of a source \mathcal{S} is defined on the same alphabet $\Sigma := [0 \dots r - 1]$. However, this alphabet is ordered with the symmetric order; namely, we consider the mapping $\Sigma \rightarrow \Sigma$ which maps the symbol σ to the symbol $\check{\sigma} = (r - 1) - \sigma$. This mapping is then extended to Σ^* using concatenation: for any $w \in \Sigma^*$, of the form $w = w_1 \cdot w_2 \dots w_k$, the word \check{w} is defined as $\check{w} = \check{w}_1 \cdot \check{w}_2 \dots \check{w}_k$. This is further extended to $\Sigma^{\mathbb{N}}$ in a similar way, and gives rise to a parametrization \check{M} for the source $\check{\mathcal{S}}$ which is closely related to the initial parametrization of the source \mathcal{S} via the equality $\check{M}(u) = M(1 - u)$.

2.4.3 Geometry of the source

This parametrization will play a central role in our work. It allows to consider a probabilistic source as a random variable defined on the unit interval. Each (infinite) word X is represented by its parameter u for which the equality $X = M(u)$ holds.

Fundamental intervals. We first define the fundamental intervals, related (as previously in the case of a dynamical source) to possible finite prefixes.

Definition 2.17. For $w \in \Sigma^*$, the interval $\mathcal{I}_w := [a_w, b_w]$ is called the fundamental interval relative to the prefix w . It gathers (up to a denumerable set) all the real numbers u for which $M(u)$ begins with the finite prefix w . It is of length p_w . When $w \in \Sigma^k$, it is said of depth k .

Parametrized version of the coincidence. We recall that we are interested in a more realistic cost related to the number of symbol comparisons needed to distinguish two words, and closely related to the coincidence, defined in Section 2.1. We now describe a parametrized version of the coincidence.

Definition 2.18. *The coincidence $\gamma(u, t)$ is the length of the largest common prefix of words $M(u)$ and $M(t)$. The function γ is then defined on the square $[0, 1] \times [0, 1]$ and associates to the pair (u, t) the coincidence $\gamma(u, t) := c(M(u), M(t))$. The realistic cost of the comparison between $M(u)$ and $M(t)$ then equals $\gamma(u, t) + 1$*

Fundamental triangles. A pair of words of the source is of the form $(M(u), M(t))$, with $(u, t) \in [0, 1]^2$. All the functions of interest only depends on the set $\{M(u), M(t)\}$. Thus, it is convenient to deal with the unit triangle

$$\mathcal{T} = \{(u, t), 0 \leq u \leq t \leq 1\}, \quad (2.15)$$

which is the set of parameters needed to describe the ordered pairs of words $(M(u), M(t))$ with $u \leq t$. We now present the notion of fundamental triangle that gathers and “mixes” the two notions that we have already introduced – fundamental interval \mathcal{I}_w and coincidence function γ .

Definition 2.19. *Consider the unit triangle \mathcal{T} defined in (2.15). For $w \in \Sigma^*$, the fundamental triangle \mathcal{T}_w associated to the prefix w is the set of all the pairs $(u, t) \in \mathcal{T}$ for which the two words $M(u)$ and $M(t)$ both begin with the prefix w , namely*

$$\mathcal{T}_w := (\mathcal{I}_w \times \mathcal{I}_w) \cap \mathcal{T} = \{(u, t), a_w \leq u \leq t \leq b_w\}. \quad (2.16)$$

Figure 2.9 represents the family of triangles \mathcal{T}_w , which defines the “geometry” of the source for two memoryless sources that are built on an alphabet of cardinality 2 and 3 respectively. The sum of all the triangle areas involves the Λ series defined in (2.1), under the form

$$\frac{1}{2} \sum_{w \in \Sigma^*} p_w^2 = \frac{1}{2} \Lambda(2),$$

and equals $(1/2)c(\mathcal{S})$ where $c(\mathcal{S})$ is the coincidence of the source, already mentioned in Section 2.1.

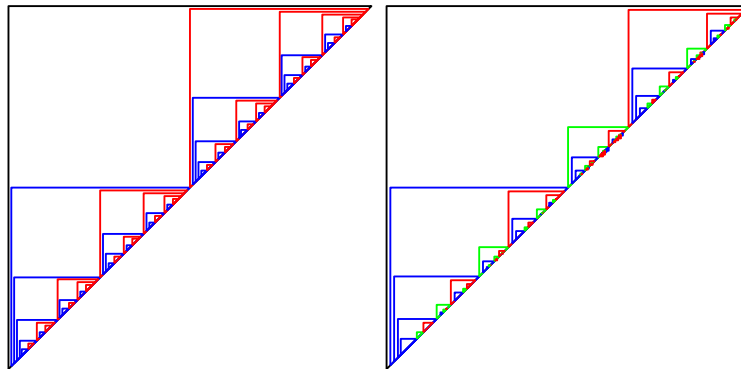


Figure 2.9: The geometry of two memoryless sources. On the left, the case of $\Sigma := \{a, b\}$ with $p_a = p_b = 1/2$. On the right, the case of $\Sigma := \{a, b, c\}$ with $p_a = 1/2, p_b = 1/6, p_c = 1/3$

Level sets of the coincidence function. The level sets of the function γ , namely the sets $[\gamma \geq \ell + 1] \cap \mathcal{T}$ are written as a disjoint union of fundamental triangles \mathcal{T}_w . More precisely, the equality

$$[\gamma \geq \ell + 1] \cap \mathcal{T} = \bigcup_{w \in \Sigma^\ell} \mathcal{T}_w$$

expresses that the coincidence $\gamma(u, t)$ is at least $\ell + 1$ if and only if the two words $M(u)$ and $M(t)$ begin by the same prefix of length ℓ . Then, the two relations

$$\mathcal{T} \cap [\gamma \geq \ell + 1] = \bigcup_{w \in \Sigma^\ell} \mathcal{T}_w, \quad \sum_{\ell \geq 0} \mathbf{1}_{[\gamma \geq \ell]} = \sum_{\ell \geq 0} (\ell + 1) \mathbf{1}_{[\gamma = \ell]},$$

entail the following equality which holds for any integrable function g on the unit triangle \mathcal{T} , and will be extensively used in the sequel,

$$\int_{\mathcal{T}} [\gamma(u, t) + 1] g(u, t) du dt = \sum_{w \in \Sigma^\star} \int_{\mathcal{T}_w} g(u, t) du dt. \quad (2.17)$$

Conclusion of the chapter. In this chapter, we have introduced different types of simple sources, we have described dynamical sources and focused on the general model of sources we introduce for our further analyses: the *parametrized* sources. We associate with each type of sources its Λ -series, which can be viewed as a generating function of the source. This series plays a center role in the analysis of the algorithms which deal with words, and this will be also the case here, in the thesis. The analytic properties of the Λ series encapsulates the probabilistic properties of the source and “returns” them in our further analyses. This is why the following chapter 3 is devoted to study the analytic properties of the Λ series of the source.

Chapter 3

Tameness of Sources

Contents

3.1	Properties on the line $\{\Re s = 1\}$. (Quasi-)periodicity.	54
3.1.1	Case of a memoryless source.	54
3.1.2	Case of a Markov chain.	57
3.1.3	Case of a dynamical source.	58
3.1.4	Quasi periodicity is exceptional.	59
3.2	Properties on the left half-plane $\{\Re s < 1\}$. Tameness.	60
3.2.1	Shape of tameness regions.	60
3.2.2	Three types of tameness	60
3.2.3	Tameness in vertical strips “with holes”. The periodic case.	61
3.2.4	Tameness in vertical strips. The UNI condition.	61
3.2.5	Tameness in hyperbolic regions. Diophantine conditions.	62
3.2.6	A small piece of history.	64
3.2.7	Conclusion of the study for classical sources.	64
3.3	Weak tameness of source.	64

The methodology of analytic combinatorics relies on the study of analytical properties of generating functions. And here, our main generating function is the Λ series, which is the Dirichlet generating function of the fundamental probabilities. We now consider the generating function $\Lambda : s \mapsto \Lambda(s)$ as a function of the complex variable s , and explain how probabilistic properties of the source may be transferred into analytic properties of this Λ function.

This Λ series will intervene as a main object when we use the Rice method. As we will see in Chapters 5, 6 and 7, we need to locate regions of the complex plane where the Λ series fulfills two properties: it is analytic (or meromorphic, with a good knowledge of the positions of its poles), and of polynomial growth when $|\Im s|$ becomes large.

However, there are other Dirichlet series, which will intervene in our analyses, as a characteristic object of the pair “source, algorithm”. They are called the mixed Dirichlet, and they will be related to the series Λ of the source (sometimes to the other generating function Π defined in Definition 2.1). As we wish to use the Rice method on these mixed Dirichlet series, we are interested in their tameness, which is itself related to the tameness of the Λ series. The mixed Dirichlet depends on the type of the algorithm (sorting or searching algorithms): for sorting algorithms (see Chapter 6), the mixed Dirichlet series are closed related to the Λ series, we are interested in the tameness of Λ on the left of the

vertical line $\Re s = 1$, and we ask there “strong” properties for Λ . For searching algorithms (see Chapter 7), the mixed Dirichlet can be also related to the Λ series, and sometimes to the Π series ; we are interested in the tameness of Λ on the right of the vertical line $\Re s = 1$, and Π on the right of the vertical line $\Re s = 0$ and we ask there “weak ” properties for Λ or for Π .

Plan of the chapter. This Chapter contains three sections; the first two are devoted to study strong properties of $\Lambda(s)$ on the left of $\Re s = 1$, which will be used in the analysis of sorting algorithms, whereas the third section reviews weak properties of $\Lambda(s)$ on the right of $\Re s = 1$, or $\Pi(s)$ on the right of $\Re s = 0$ which will be used in the analysis of searching algorithms

For our study of strong analytical properties of $\Lambda(s)$, as we already know that the series $\Lambda(s)$ is not defined at $s = 1$, our study will be divided into two steps. We first consider the vertical line $\{\Re s = 1\}$, and then the situation on the left of this vertical line. We are also interested in the behaviour of the bivariate series $\Lambda(s, u)$ when $\Re s$ is close to 1 and u close to 1. We first describe the properties of the Λ series for classical sources that are considered in the previous chapter (memoryless sources, Markov chains, dynamical sources). The study of these particular cases gives rise to interesting phenomena: periodicity, quasi-periodicity, and tameness. It is thus natural to propose (at the end of the chapter) a classification of general (non-ambiguous) sources with respect to these phenomena.

In the first section of this chapter, we study the properties of the Λ -series of our “classical” sources (memoryless sources, Markov chains, dynamical sources) on the half-plane $\{\Re s \geq 1\}$ and we study two important phenomena which characterise the behaviour of the Λ series on the vertical line $\{\Re s = 1\}$, namely the periodicity¹ and the quasi-periodicity.

In the second section of this chapter, we describe the properties of the Λ -series of our “classical” sources (memoryless sources, Markov chains, dynamical sources) on the half-plane $\{\Re s < 1\}$ and we study an important phenomenon which describes the situation of the Λ series on this half-plane, namely the shape of their tameness regions. Finally, we will give general definitions for periodicity, quasi-periodicity and tameness for general sources.

Finally, the last section is devoted to tameness properties of another type, which describe weaker properties of $\Lambda(s)$ on the right of $\Re s = 1$, and properties of the series $\Pi(s)$ on the right of $\Re s = 0$. These properties will be sufficient to entail tameness of the Dirichlet series which intervene in the analysis of searching algorithms, in Chapter 7

3.1 Properties on the line $\{\Re s = 1\}$. (Quasi-)periodicity.

3.1.1 Case of a memoryless source.

We begin with this simplest case, where we use the expression of the Λ series previously obtained in Lemma 2.7:

$$\Lambda(s) = \frac{1}{1 - \lambda(s)}.$$

Characterization of the set of poles. The following result describes the first properties of the Λ series.

¹We recall that the periodicity phenomenon (which may also occur in the Markov chain) must not be mixed up with the usual aperiodicity property for Markov chains which we never used under this terminology: in this thesis, a Markov chain which is irreducible and aperiodic (in the usual sense) will be said to be good.

Lemma 3.1. *The Dirichlet series $\Lambda(s)$ of a memoryless source is meromorphic on the complex plane, is analytic on the half plane $\Re s > 1$ and has a simple pole at $s = 1$. Moreover, the set \mathcal{Z} of poles is defined as:*

$$\mathcal{Z} = \{s; \lambda(s) = 1\}. \quad (3.1)$$

Proof. The function $s \mapsto \lambda(s)$ defined in 2.1 is analytic on the complex plane, and thus the function $s \mapsto \Lambda(s)$ is meromorphic with a set of poles \mathcal{Z} defined in (3.1). Let $\sigma := \Re s$, and assume $\sigma > 1$. Then, the inequality $|\lambda(s)| \leq \lambda(\sigma) < \lambda(1) = 1$ entails that the set \mathcal{Z} is contained in the half plane $\Re s \leq 1$. \square

Periodicity. To the family of probabilities $\mathfrak{P} = (p_1, p_2, \dots, p_r)$, we associate the ratios

$$\alpha(i, j) := \frac{\log p_i}{\log p_j} \quad \text{for any pair } (i, j) \in \Sigma^2. \quad (3.2)$$

The following classical result proves that the position of the set \mathcal{Z} with respect to the vertical line $\Re s = 1$ is related to the rationality of the ratios $\alpha(i, j)$ (see [26]).

Lemma 3.2. *For a memoryless source of probabilities \mathfrak{P} , the following conditions are equivalent:*

- (a) *The intersection $\mathcal{Z} \cap \{\Re s = 1\}$ contains a point $s \neq 1$.*
- (b) *All the ratios $\alpha(i, j)$ defined in (3.2) are rational numbers.*
- (c) *There exists $\tau > 0$ for which the equality $\mathcal{Z} \cap \{\Re s = 1\} = 1 + 2i\pi\tau\mathbb{Z}$ holds.*
- (d) *The function $\lambda(s)$ is periodic of period $2i\pi\tau$.*

A source which satisfies one of these conditions is said to be periodic.

When a memoryless source is periodic, then (e) holds

(e) *there exists an algebraic integer $a < 1$ for which all the probabilities p_i belong to the semi-group generated by a .*

Example. For any unbiased memoryless source, all the ratios $\alpha(i, j)$ are equal to 1. Such a source is periodic, and $\lambda(s)$ is periodic of period $(2\pi)i/\log r$. An instance of a (non trivial) periodic memoryless source on the binary alphabet is given by $p_0 = 1/\phi, p_1 = 1/\phi^2$ where $\phi = (1 + \sqrt{5})/2$ is the golden ratio.

Proof. We will prove $(a) \Rightarrow (b) \Rightarrow (c) \Rightarrow (d) \Rightarrow (a)$

$(a) \Rightarrow (b)$. For $s = 1 + 2i\pi t$, one has

$$\lambda(s) = \sum_{\ell \in \Sigma} p_\ell e^{2i\pi t \log p_\ell} \quad \text{so that} \quad |\lambda(s)| = \left| \sum_{\ell \in \Sigma} p_\ell e^{2i\pi t \log p_\ell} \right| \leq \sum_{\ell \in \Sigma} p_\ell = 1.$$

Now, if the equality $\lambda(1 + 2i\pi t) = 1$ holds, this entails (by the converse of the triangular inequality) the following equalities,

$$\forall \ell, \quad e^{2i\pi t \log p_\ell} = 1, \quad \text{and then} \quad t \log p_\ell \in \mathbb{Z}. \quad (3.3)$$

Now, if $t \neq 0$, one deduces

$$\alpha(i, j) := \frac{\log p_i}{\log p_j} \in \mathbb{Q}, \quad \forall i, j.$$

(b) \Rightarrow (c). The previous proof shows that the possible solutions of the equation $\lambda(1+2i\pi t) = 1$ arise when

$$t \in \frac{1}{\log p_j} \mathbb{Z} \quad \forall j \in [1 \dots r].$$

When all the real numbers $\alpha(i, j)$ are rational, the intersection of the lattices generated by $(1/\log p_i)$ is not reduced to $\{0\}$ and has a smallest element $\tau > 0$. Then, all the solutions of the equation $\lambda(1+2i\pi t) = 1$ are of the form $t = \tau \mathbb{Z}$.

(c) \Rightarrow (d). One always has, for any real t ,

$$\lambda(s + 2i\pi t) = \sum_{\ell} p_{\ell}^{s+2i\pi t} = \sum_{\ell} p_{\ell}^s e^{2i\pi t \log p_{\ell}}.$$

Assume now that the equality $\lambda(1+2i\pi t) = 1$ holds with $t \in \tau \mathbb{Z}$. Then, the proof of (a) \Rightarrow (b) shows that Relation (3.3) holds and then, for any $t \in \tau \mathbb{Z}$, one has

$$\lambda(s + 2i\pi t) = \sum_{\ell} p_{\ell}^s e^{2i\pi t \log p_{\ell}} = \sum_{\ell} p_{\ell}^s = \lambda(s).$$

(d) \Rightarrow (a). Clear.

(d) \Rightarrow (e). Clear if we let $a := \exp(1/\tau)$ □

Quasi-periodicity. For a general complex number of modulus 1, we will be also interested in the description of the set

$$\underline{\mathcal{Z}} := \{s; |\lambda(s)| = 1\},$$

and its position with respect to the vertical line $\{\Re s = 1\}$. The following result provides an extension of the previous Lemma 3.2. It is related to the rationality of the ratios $\alpha(k, j, \ell)$, defined as

$$\alpha(i, j, k) := \frac{\log p_i - \log p_j}{\log p_i - \log p_k} \quad \text{for any pair } (i, j, k) \in \Sigma^3. \quad (3.4)$$

Lemma 3.3. *For a memoryless source of probabilities \mathfrak{P} , the following conditions are equivalent:*

- (a) *The intersection $\underline{\mathcal{Z}} \cap \{\Re s = 1\}$ contains a point $s \neq 1$.*
- (b) *All the ratios $\alpha(i, j, k)$ defined in (3.4) are rational numbers.*
- (c) *There exists $\tau > 0$ for which the equality $\underline{\mathcal{Z}} \cap \{\Re s = 1\} = 1 + i\tau \mathbb{Z}$ holds.*

A source which satisfies one of these conditions is said to be quasi-periodic.

Proof. We will prove (a) \Rightarrow (b) \Rightarrow (c)

(a) \Rightarrow (b). For $s = 1 + 2i\pi t$, one has

$$\lambda(s) = \sum_{\ell \in \Sigma} p_{\ell} e^{2i\pi t \log p_{\ell}} \quad \text{so that} \quad |\lambda(s)| = \left| \sum_{\ell \in \Sigma} p_{\ell} e^{2i\pi t \log p_{\ell}} \right| \leq \sum_{\ell \in \Sigma} p_{\ell} = 1.$$

Now, if the equality $\lambda(1+2i\pi t) = u = e^{2i\pi\theta}$ holds, this entails (by the converse of the triangular inequality) the following equalities,

$$\forall \ell, \quad e^{2i\pi t \log p_{\ell}} = u = e^{2i\pi\theta}, \quad \text{and then} \quad t \log p_{\ell} \in \theta + \mathbb{Z}, \quad t(\log p_i - \log p_j) \in \mathbb{Z}. \quad (3.5)$$

Thus, if $t \neq 0$, one deduces

$$\alpha(i, j, k) := \frac{\log p_i - \log p_j}{\log p_i - \log p_k} \in \mathbb{Q} \quad \forall i, j, k.$$

(b) \Rightarrow (c). The previous proof shows that the possible solutions of the equation $\lambda(1+2i\pi t) = u$ verify

$$t \in \frac{1}{\log p_i - \log p_j} \mathbb{Z}.$$

When all the real numbers $\alpha(i, j, k)$ are rational, the intersection of all these lattices is a not reduced to $\{0\}$ and is of the form $\mathbb{Z}\tau$, with $\tau > 0$.

Fix an element of the lattice $\mathbb{Z}\tau$, of the form $t = k\tau$ for $k \in \mathbb{Z}$, and denote by θ_k the real $\theta_k \in [0, 2\pi[$ for which $k\tau \log p_1 \equiv \theta_k \pmod{1}$. Now, the pair $(k\tau, \theta_k)$ is a solution of the system

$$k\tau \log p_1 \equiv \theta \pmod{1}, \quad k\tau \log p_2 \equiv \theta \pmod{1}, \dots, \quad k\tau \log p_r \equiv \theta \pmod{1}.$$

This means that the set $\underline{\mathcal{Z}} \cap \{\Re s = 1\}$ coincides with the set $\{1 + 2i\pi k\tau, \quad k \in \mathbb{Z}\}$. □

3.1.2 Case of a Markov chain.

We now consider the case of a good Markov chain, where the expression of Λ is provided in Lemma 2.10.

Characterization of the set of poles. There is an analog of Lemma 3.1 in the case of a good Markov chain.

Lemma 3.4. *The Dirichlet series $\Lambda(s)$ of a good Markov chain is meromorphic on the complex plane, is analytic on the half plane $\Re s > 1$ and has a simple pole at $s = 1$. Moreover, the set \mathcal{Z} of poles is defined as*

$$\mathcal{Z} = \{s; \quad \det(I - \mathbf{P}_s) = 0\}. \quad (3.6)$$

Proof. The function $s \mapsto \mathbf{P}_s$ is analytic on the complex plane, and thus the function $s \mapsto \Lambda(s)$ is meromorphic with a set of poles \mathcal{Z} defined in (3.6). Let $\sigma := \Re s$. Then, the inequality $\|\mathbf{P}_s^k\| \leq \|\mathbf{P}_\sigma^k\|$ holds and entails the inequality on the spectral radius $r(s) \leq r(\sigma)$. In the case of a good Markov chain, the spectral radius $r(\sigma)$ equals the dominant eigenvalue $\lambda(\sigma)$. Assume now the strict inequality $\sigma > 1$, we wish to prove the strict inequality $\lambda(\sigma) < \lambda(1) = 1$. As the inequality $\lambda(\sigma) \leq \lambda(1)$ holds, we assume that the equality $\lambda(\sigma) = \lambda(1)$ holds, and we look for a contradiction.

The equalities

$$\sum_j p_{i|j}^\sigma \pi_\sigma^{(j)} = \lambda(\sigma) \pi_\sigma^{(i)}, \quad \lambda(1) = 1 = \sum_i p_{i|j} = \sum_j \pi_\sigma^{(j)}$$

entail

$$\lambda(\sigma) = \sum_{i,j} p_{i|j}^\sigma \pi_\sigma^{(j)} = \sum_j \pi_\sigma^{(j)} \sum_i p_{i|j}^\sigma,$$

and thus

$$0 = \lambda(1) - \lambda(\sigma) = \sum_j \pi_\sigma^{(j)} \left[\sum_i (p_{i|j} - p_{i|j}^\sigma) \right].$$

This implies that for any $i \in \Sigma$ there is a unique $j = \tau(i) \in \Sigma$ for which the probability $p_{i|j} = 1$. When the Markov chain is good, there does not exist such a map $\tau : \Sigma \rightarrow \Sigma$. □

Periodicity. Consider a good Markov chain, its transition matrix \mathbf{P} , and, for any cycle of length $k \geq 1$, of the form $\mathcal{C} := \{i_1 i_2 \dots i_k\}$, its probability $p(\mathcal{C}) := p_{i_1|i_k} p_{i_2|i_1} \dots p_{i_k|i_{k-1}}$ and its normalized probability $\pi(\mathcal{C}) = p(\mathcal{C})^{1/k}$. We also consider all the possible ratios of the form

$$\alpha(\mathcal{C}, \mathcal{K}) := \frac{\log \pi(\mathcal{C})}{\log \pi(\mathcal{K})} \quad \text{for each pair } (\mathcal{C}, \mathcal{K}) \text{ of cycles,} \quad (3.7)$$

$$\alpha(\mathcal{C}, \mathcal{K}, \mathcal{L}) := \frac{\log \pi(\mathcal{C}) - \log \pi(\mathcal{K})}{\log \pi(\mathcal{C}) - \log \pi(\mathcal{L})} \quad \text{for each triple } (\mathcal{C}, \mathcal{K}, \mathcal{L}) \text{ of cycles.} \quad (3.8)$$

These ratios play a similar role to the previous ratios $\alpha(i, j)$ or $\alpha(i, j, k)$ in the memoryless case. Indeed, the following result holds and extends the previous Lemmas 3.2 and 3.3. Its proof is omitted.

Lemma 3.5. *For a good Markov chain, with transition matrix \mathbf{P} , the following conditions are equivalent:*

- (a) *The intersection $\mathcal{Z} \cap \{\Re s = 1\}$ contains a point $s \neq 1$.*
- (b) *All the ratios $\alpha(\mathcal{K}, \mathcal{L})$ defined in (3.7) are rational.*
- (c) *There exists $\tau > 0$ for which the equality $\mathcal{Z} \cap \{\Re s = 1\} = 1 + 2i\pi\tau\mathbb{Z}$ holds.*
- (d) *The matrix $s \mapsto \mathbf{P}_s$ is periodic of period $i\tau$.*

A Markov chain which satisfies one of these conditions is said to be periodic.

When a Markov chain is periodic, there exists an algebraic integer a and a vector of positive reals $(\nu_1, \nu_2, \dots, \nu_r)$ for which the matrix \mathbf{P} is written as $\mathbf{P} = D^{-1}\mathbf{Q}D$, where D is the matrix whose diagonal is $(\nu_1, \nu_2, \dots, \nu_r)$ and all the nonzero coefficients of the matrix \mathbf{Q} belong to the group generated by a .

Quasi-periodicity. We will be also interested in the description of the set

$$\underline{\mathcal{Z}} := \{s; \exists u, |u| = 1, \det(I - u\mathbf{P}_s) = 0\},$$

and its position with respect to the vertical line $\{\Re s = 1\}$. The following result provides an extension of the previous Lemma 3.5 in the case when u is a general complex of modulus 1. It is related to the rationality of the ratios $\alpha(\mathcal{C}, \mathcal{K}, \mathcal{L})$, defined in (3.8).

Lemma 3.6. *For a good Markov chain, with transition matrix \mathbf{P} , the following conditions are equivalent:*

- (a) *The intersection $\underline{\mathcal{Z}} \cap \{\Re s = 1\}$ contains a point $s \neq 1$.*
- (b) *All the ratios $\alpha(\mathcal{C}, \mathcal{K}, \mathcal{L})$ defined in (3.8) are rational.*
- (c) *There exists $\tau > 0$ for which the equality $\underline{\mathcal{Z}} \cap \{\Re s = 1\} = 1 + 2i\pi\tau\mathbb{Z}$ holds.*

A Markov chain which satisfies one of these conditions is said to be quasi-periodic.

3.1.3 Case of a dynamical source.

We now consider the case of a dynamical system, where Lemma 2.13 provides an expression of the generating function $\Lambda(s)$ as a function of the transfer operator \mathbb{H}_s defined in (2.9). Moreover, we limit ourselves to good dynamical sources whose definition is given in Definition 2.14. We now recall the statements of Lemma 3.7 in a more concise form.

Lemma 3.7. *In the case of a good dynamical source, the function $\Lambda(s)$ is analytic on the half plane $\{\Re s > 1\}$ and admits a simple pole at $s = 1$.*

Now, we study the possible periodicity (or quasi-periodicity) of such a good dynamical source, and we introduce the analogs of quantities $\alpha(i, j)$ defined in (3.2) for memoryless sources or $\alpha(\mathcal{C}, \mathcal{K})$ defined in (3.7) for Markov chains. All these quantities are defined with cycles, and we are then led to study the fixed points of inverse branches $h \in \mathcal{H}^*$. First, it is clear that, for a good dynamical system, any inverse branch $h \in \mathcal{H}^*$ has a unique fixed point, denoted by h^* .

For an inverse branch $h \in \mathcal{H}^*$, we denote the depth of h by $p(h)$ and for $h, k, \ell \in \mathcal{H}^*$, we consider

$$\pi(h) := |h'(h^*)|^{1/p(h)}, \quad \alpha(h, k) := \frac{\log \pi(h)}{\log \pi(k)}, \quad \alpha(h, k, \ell) := \frac{\pi(h) - \pi(k)}{\pi(h) - \pi(\ell)}. \quad (3.9)$$

These ratios $\alpha(h, k)$ or $\alpha(h, k, \ell)$ provide an extension of our previous quantities α that we have already defined for simple sources. The following result is an extension of the previous results described in Lemma 3.2, 3.3, 3.5 and 3.6. It relates the possible singularities of the quasi-inverse $(I - u\mathbb{H}_s)^{-1}$ on the vertical line $\Re s = 1$ to the rationality of the ratios α 's. However, as there is an infinite number of possible ratios α , there are no longer exact equivalences.

Proposition 3.8. *Consider a dynamical system of the Good Class and its secant transfer operator \mathbb{H}_s , acting on the space $\mathcal{C}^1(\mathcal{I} \times \mathcal{I})$. Then, the following holds:*

- (a) *If there exists $t_0 \neq 0$ for which the spectrum $\text{Sp } \mathbb{H}_{1+it_0}$ contains an eigenvalue equal to 1, then all the ratios $\alpha(h, k)$ are rational numbers, and the set of the real numbers t for which the spectrum $\text{Sp } \mathbb{H}_{1+it}$ contains an eigenvalue equal to 1 is a lattice $\mathbb{Z} \cdot \tau$ for some $\tau > 0$. In this case, the source is said to be periodic.*
- (b) *If there exists a ratio $\alpha(h, k)$ which is not rational, then, the quasi-inverse $(I - \mathbb{H}_s)^{-1}$ is analytic on $\Re s = 1$ except at $s = 1$ where it has a simple pole.*
- (c) *If there exists $t_0 \neq 0$ for which the spectrum $\text{Sp } \mathbb{H}_{1+it_0}$ contains an eigenvalue u of modulus 1, then all the ratios $\alpha(h, k, \ell)$ are rational numbers, and the set of the real numbers t for which the spectrum $\text{Sp } \mathbb{H}_{1+it}$ contains eigenvalue u of modulus 1 is a lattice $\mathbb{Z} \cdot \tau$ for some $\tau > 0$. In this case, the source is said to be quasi-periodic.*
- (d) *If there exists a ratio $\alpha(h, k, \ell)$ which is not rational, then, the spectral radius of \mathbb{H}_s is strictly less than 1 on $\{s; \Re s = 1, s \neq 1\}$ and, for any u of modulus 1, the quasi-inverse $(I - u\mathbb{H}_s)^{-1}$ is analytic on the line $\Re s = 1$ except for $s = 1$ and $u = 1$ where it admits a simple pole.*

3.1.4 Quasi periodicity is exceptional.

We have mentioned that the periodicity phenomenon arises in a natural context for simple sources, since any unbiased memoryless source is periodic. Then a natural further question is: *Do there exist many general dynamical sources which are periodic? quasi-periodic?*

The following result shows that the (quasi)-periodicity phenomenon is in a sense exceptional for general good dynamical sources: It only occurs for sources which are obtained from simple sources by some conjugation. We recall that two dynamical sources (\mathcal{I}, T) and (\mathcal{I}, U) are conjugate if there exists a bijection $\Phi : \mathcal{I} \rightarrow \mathcal{I}$ of class \mathcal{C}^2 for which $T : \Phi \circ U \circ \Phi^{-1}$. Then, it is clear that a dynamical system conjugate to a periodic dynamical system is itself periodic. More precisely, it is proven for instance in [4] the following:

Proposition 3.9. *A (complete) good dynamical source may be quasi-periodic only if it is conjugate to a source with affine branches.*

We will be more interested in the contrapositive of the previous statement:

\mathcal{NA} . Non-Affine. If a good dynamical system is not conjugate to a dynamical system with affine branches, then $\Lambda(s, u)$ is analytic for any $(s, u) \neq (1, 1)$, with $\Re s = 1, |u| = 1$.

3.2 Properties on the left half-plane $\{\Re s < 1\}$. Tameness.

The previous section describes the position of the set \mathcal{Z} of singularities of the Λ -function in the half-plane $\{\Re s \geq 1\}$. We now focus on the left halfplane $\{\Re s < 1\}$ and isolate a region $\mathcal{R} \supset \{\Re s < 1\}$ where the Λ function is analytic. In fact, we have to re-inforce our needs for the region \mathcal{R} , because it is also essential (for applying the Rice method, as we shall see it later in Chapter 4) that $\Lambda(s)$ be of polynomial growth for $s \in \mathcal{R}, |\Im s| \rightarrow \infty$. Such a region will play a central role in the subsequent analyses. We are then led to the following definition.

Definition 3.10. [Tameness region] A tameness region for a source \mathcal{S} is a region $\mathcal{R} \supset \{\Re s \geq 1\}$ where the Λ series is meromorphic, with a only pole (simple) located at $s = 1$, and is of polynomial growth when $|\Im s| \rightarrow \infty$.

We wish to obtain sufficient conditions on probabilistic probabilities of the source under which the “shape” of a tameness region \mathcal{R} for \mathcal{S} may be made precise.

3.2.1 Shape of tameness regions.

As we will see later, regions with an hyperbolic shape or vertical strips arise in a natural way as possible tameness regions for interesting subclasses of sources. This is why the following definition will be important in the sequel. It describes three cases of possible tameness regions \mathcal{R} which will occur for a classical source:

Definition 3.11. [Shape of regions] A region $\mathcal{R} \supset \{\Re s \geq 1\}$ has a

- (a) *S-shape* (shorthand for Strip shape) if \mathcal{R} is a vertical strip $\Re(s) > 1 - \delta$ for some $\delta > 0$.
- (b) *H-shape* (shorthand for Hyperbolic shape) if \mathcal{R} is an hyperbolic region \mathcal{R} , defined as, for some $A, B, \rho > 0$

$$\mathcal{R} := \{s = \sigma + it; \quad |t| \geq B, \quad \sigma > 1 - \frac{A}{|t|^\rho}\} \cup \{s = \sigma + it; \quad \sigma > 1 - \frac{A}{B^\rho}, |t| \leq B\}.$$

- (c) *P-shape* (shorthand for Periodic shape) if \mathcal{R} is a vertical strip “with holes”, namely

$$\mathcal{R} := \mathcal{R}_0 \setminus \mathcal{R}_1, \quad \mathcal{R}_0 := \{\Re s > 1 - \delta\}, \quad \mathcal{R}_1 := \{s = 1 + it; \quad t = 2i\pi k\tau, k \in \mathbb{Z} \setminus \{0\}\},$$

for some $\delta, \tau > 0$.

When they exist, δ is the width, ρ is the hyperbolicity exponent, and τ is the period.

3.2.2 Three types of tameness

Definition 3.12. [Shape of tameness]. A source is Λ -tame if its Λ -series satisfies the following:

- (a) It admits at $s = 1$ a simple pole, with a residue equal to $1/h(\mathcal{S})$, (where $h(\mathcal{S})$ is the entropy of the source)
- (b) It admits a tameness region with one of the shapes that are described in Definition 3.11.

A vertical strip can be viewed as a region with a zero hyperbolicity exponent. We are interested by tameness regions which are the largest possible. Then, it is natural to define the hyperbolicity exponent of the source \mathcal{S} as the *infimum of all the hyperbolicity exponents* of tameness regions of the source \mathcal{S} . For instance, if the source admits as tameness region a vertical strip, then the hyperbolicity exponent of the source equals 0. There also exist some sources for which the singularities of the Λ function come close to the vertical line $\Re s = 1$ very fast, with an exponential speed. Such sources have an hyperbolicity exponent equal to ∞ .

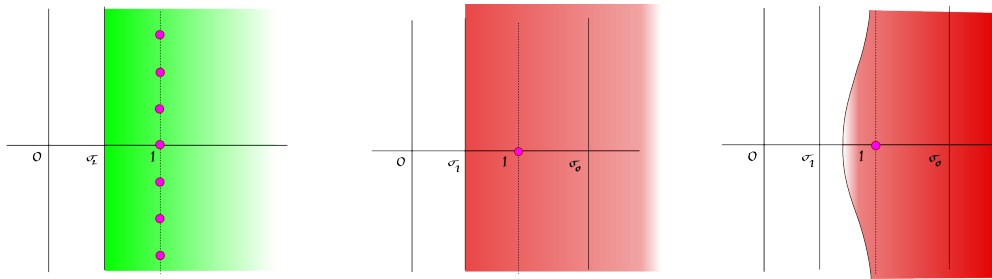


Figure 3.1: Three possible tameness regions \mathcal{R} for the Λ series. On the left, the case of a periodic source. On the middle, the case of the S -shape. On the right, the case of a H -shape.

3.2.3 Tameness in vertical strips “with holes”. The periodic case.

In the case of a periodic source, which is conjugate to a simple source, the function $s \mapsto \lambda(s)$ is periodic of period $i\tau$, and there is a vertical strip on the left of the vertical line $\Re s = 1$ where the Λ function is analytic and of polynomial growth. There exists in this case a tameness region of the source which is a “vertical strip with holes”. Then:

Lemma 3.13. *A periodic source is P -tame.*

3.2.4 Tameness in vertical strips. The UNI condition.

When a good dynamical source is not conjugate to a dynamical source with affine branches, we already know, with Section 3.1.4, that it cannot be periodic. Moreover, the series Λ is meromorphic on $\Re s \geq 1$, with a only pole at $s = 1$. A source which is tame with a strip shape strongly differs from a periodic source. It is thus natural to search S -tame sources amongst sources whose geometry strongly differs from the geometry of simple sources.

This is the idea of Dolgopyat when he introduces the UNI Condition that expresses that the source strongly differs from a source with affine branches.

Condition UNI. One first defines a probability Π_n on each set $\mathcal{H}^n \times \mathcal{H}^n$, in a natural way, and lets $\Pi_n\{(h, k)\} := |h(\mathcal{I})| \cdot |k(\mathcal{I})|$, where $|\mathcal{J}|$ denotes the length of the interval \mathcal{J} .

Furthermore, $\Delta(h, k)$ denotes the “distance” between two inverse branches h and k of same depth, defined as

$$\Delta(h, k) = \inf_{x \in \mathcal{I}} |\Psi'_{h,k}(x)| \quad \text{with} \quad \Psi_{h,k}(x) = \log \left| \frac{h'(x)}{k'(x)} \right|. \quad (3.10)$$

The distance $\Delta(h, k)$ is a measure of the difference between the “shape” of the two branches h, k . The UNI Condition, stated as follows [20], is a geometric condition which expresses that the probability that two inverse branches have almost the same “shape” is very small:

Definition 3.14. [Condition UNI]. *A good dynamical system (\mathcal{I}, T) with contraction ratio $\rho < 1$ satisfies the UNI condition if its set \mathcal{H}^* of inverse branches satisfies the following*
For any $\hat{\rho} \in]\rho, 1[$, and for any integer n , one has $\Pi_n[\Delta \leq \hat{\rho}^n] \ll \hat{\rho}^n$.

For a source with affine branches, the “distance” Δ is always zero, and the probabilities $\Pr_n[\Delta \leq \hat{\rho}^n]$ are all equal to 1. Such a source never satisfies the Condition UNI. Conversely, a good dynamical source of the UNI Class cannot be conjugate to a source with affine branches, as it is proven by Baladi and Vallée [5]. Then, the condition UNI excludes all the simple sources, which cannot be S-tame.

Condition UNI and S-tameness. The strength of the Condition UNI is due to the fact that this condition is sufficient to imply S-tameness:

Theorem 3.15. [Dolgopyat, Baladi–Vallée, Cesaratto–Vallée] *A good dynamical system that satisfies the condition UNI is S-tame.*

3.2.5 Tameness in hyperbolic regions. Diophantine conditions.

There are other conditions, of arithmetical type, which are sufficient to imply H -tameness.

Irrationality exponent and diophantine conditions. The irrationality exponent of an irrational number was introduced by Liouville. The irrationality exponent of the irrational number x is defined by

$$\mu(x) := \sup \left\{ \nu, \left| x - \frac{p}{q} \right| \leq \frac{1}{q^{2+\nu}} \text{ for an infinite number of pairs } (p, q) \right\}.$$

The irrationality exponent of the irrational x is then a measure of its approximability by rational numbers.

The approximability of an irrational number x is closely related to its continued fraction expansion, since truncations of this expansion give rise to the rational numbers that provide the best rational approximations of the irrational x . When all the quotients that occur in the continued fraction expansion of x are bounded, the irrational x cannot be well approximable by rationals.

An irrational number x is diophantine if its irrationality exponent is finite. Then, a diophantine irrational number is not too well approximable by rational numbers: it can be viewed (in an informal way...) as an irrational number “quite different” from a rational number. An irrational number whose quotients that occur in its continued fraction expansion of x are bounded is diophantine.

H-tameness of simple sources. It is possible to define the irrationality exponent of a finite family of numbers, provided that they are not all rational. The irrationality $\mu(\mathcal{S})$ of a non-periodic simple source \mathcal{S} , is then defined as the irrationality exponent of the set

$$\{\alpha(\mathcal{C}, \mathcal{K}); \quad \mathcal{K}, \mathcal{C} \text{ cycles of length } \leq r\}.$$

The source is diophantine if the irrationality exponent is finite. For a memoryless source over an alphabet of size r , the irrationality exponent satisfies almost everywhere the inequality

$$\mu(\mathfrak{P}) + 1 = \frac{1}{r - 1}.$$

(“everywhere” means here: when the probability family \mathfrak{P} is randomly chosen in the subset $\{(p_1, p_2, \dots, p_r) : p_j > 0, p_1 + p_2 + \dots + p_r = 1\}$ with respect to the Lebesgue measure.)

Theorem 3.16. [Roux-Vallée] *For a simple non-periodic source, there exists an exact relation between the two exponents –the irrationality exponent μ and the hyperbolicity exponent ρ , namely the equality $\rho = 2\mu + 2$. A simple source is never S -tame. A diophantine non-periodic source is H -tame.*

The remark above together with the previous Theorem entail that the hyperbolicity exponent of a non-periodic memoryless source over an alphabet of size r is “almost everywhere” equal to $2/(r - 1)$. The hyperbolicity exponent of a binary source is “almost everywhere” equal to 2.

H-tameness of a dynamical source. The DIOP conditions. We consider a re-inforcement of Proposition 3.8. Since, from an informal point of view, a diophantine irrational number is an irrational number which is quite different from a rational number, we are interested in the following situation where the hypotheses of Proposition 3.8 are respectively re-inforced and replaced by

(b') *There exist two branches $h, k \in \mathcal{H}^*$ for which the ratio $\alpha(h, k)$ is diophantine.*

(d') *There exist three branches $h, k, \ell \in \mathcal{H}^*$ for which the ratio $\alpha(h, k, \ell)$ is diophantine.*

We remark that there is no clear relation between Hypothesis (b') and (d').

We are then led to consider the following sources:

Definition 3.17. [2DIOP and 3DIOP] *A non-periodic good dynamical source is*

- (a) *2-diophantine (2DIOP in shorthand) if there exist two branches $h, k \in \mathcal{H}^*$ for which the ratio $\alpha(h, k)$ is diophantine.*
- (b) *3-diophantine (3DIOP in shorthand) if there exist three branches $h, k, \ell \in \mathcal{H}^*$ for which the ratio $\alpha(h, k, \ell)$ is diophantine.*

The following result relates the arithmetic properties of the source probabilities to the geometry of the tameness region. This is the main contribution of Roux' PhD thesis [56].

Theorem 3.18. [Dologopyat, Naud, Melbourne, Roux-Vallée] [20][21][56] (a) *A good dynamical system that satisfies the DIOP3 condition is H -tame.*

(b) *A good dynamical system that satisfies the DIOP2 condition is H -tame.*

However, in the case of a general source, the optimality of the tameness region is not proven, and there is only an upper bound on the hyperbolicity exponent. We cannot exclude that there may exist a vertical strip as a tameness region, for which the hyperbolicity exponent equals 0. This happens when the UNI condition is also fulfilled.

3.2.6 A small piece of history.

The UNI and DIOP conditions are introduced by Dolgopyat in the paper [20][21]. He proves that, under these conditions, and in the case of a finite alphabet, the quasi inverse of the *plain* transfer operator is analytic and of polynomial growth in a region on the left of the line $\Re s = 1$. When the UNI condition holds, this region is a vertical strip. When the DIOP condition holds, this region is of hyperbolic type. There are extensions of these previous results to the quasi-inverse of the *secant* operator, which are proven to hold for the UNI condition by Cesaratto and Vallée [10][11], and for the DIOP condition by Roux and Vallée [57].

3.2.7 Conclusion of the study for classical sources.

The table of Figure 3.2 describes various possible behaviors of classical sources.

Tameness shape	Simple sources	Dynamical sources not conjugate to simple sources	Arithmetic conditions on ratios α	Geometric conditions on branches
Periodic	Possible	Impossible	$\alpha(h, k) \in \mathbb{Q}$	—
Quasi-periodic	Possible	Impossible	$\alpha(h, k, \ell) \in \mathbb{Q}$	—
Strip tame	Impossible	Possible	—	UNI
Hyperbolic tame	Possible	Possible	DIOP2 or DIOP3	—

Figure 3.2: Tameness properties for classical sources, and sufficient conditions under which these tameness properties hold

“Most of the time”, the simple sources (memoryless sources or good Markov chains) are Λ -tame. They never have a S -shape, but they may have a H -shape or a P -shape, according to arithmetic properties of their probabilities.

Dynamical sources may have a P -shape only if they are “conjugate” to simple sources. Adapting deep results of Dolgopyat [20],[21] to the “secant transfer operator” \mathbb{H}_s , it is possible to prove that dynamical sources are “most of the time” Λ -tame with a S -shape, but they may also have a H -shape. For more details, see the cited papers where all these facts, here described in a informal way, are stated in a formal way and proven.

3.3 Weak tameness of source.

We now describe tameness properties which deal with the series $\Lambda(s)$ on the right of the vertical line $\Re s = 1$. The series $\Lambda(s)$ always satisfies $\Lambda_k(1) = 1$ and thus $\Lambda(1) = +\infty$. Then, on the left of the vertical line $\Re s = 1$, the series $\Lambda(s)$ cannot be absolutely convergent, but we have deeply studied in the first two sections of this Chapter what are the possible behaviors of the series $\Lambda(s)$ there. We will see that this study will be central for sorting algorithms.

However, it will appear that such strong properties are not needed for searching algorithms, where we will be interested in the behaviour of $\Lambda(s)$ on the right of the vertical line $\Re s = 1$. The situation is completely different on the right of the vertical line, where there is an abscissa of absolute convergence $\sigma_a \geq 1$, that we now define.

Assume that there exists a real $\sigma > 1$ for which the series $\Lambda(\sigma)$ is convergent. Then, as the probabilities p_w are positive, one has $|p_w^s| = p_w^{\Re s}$ and the series $\Lambda(s)$ is convergent

on the whole vertical line $\Re s = \sigma$. Furthermore, as the probabilities p_w are at most 1, the map $\sigma \mapsto \Lambda(\sigma)$ is decreasing. Finally, if we define

$$\sigma_a(\Lambda) := \inf\{\sigma; \quad \Lambda(\sigma) < +\infty\},$$

then, the series $\Lambda(\sigma)$ is convergent for any real $\sigma > \sigma_a(\Lambda)$. Moreover, the Dirichlet series Λ is normally convergent on the halfplane $\Re s \geq \sigma$, and defines an analytic function there, which is furthermore of bounded growth on this halfplane.

The situation is similar with the series $\Pi(s)$. The series $\Pi(s)$ always satisfies $\pi_k^0 = 1$ and thus $\Pi(0) = +\infty$. Then, on the left of the vertical line $\Re s = 0$, the series $\Pi(s)$ cannot be absolutely convergent. But, on the right of the vertical line $\Re s = 0$, there is an abscissa of absolute convergence defined as

$$\sigma_a(\Pi) := \inf\{\sigma; \quad \Pi(\sigma) < +\infty\}.$$

Then, the series $\Pi(\sigma)$ is convergent for any real $\sigma > \sigma_a(\Pi)$. Moreover, the Dirichlet series Π is normally convergent on the halfplane $\Re s \geq \sigma$, and defines an analytic function there, which is furthermore of bounded growth on this halfplane.

We now give the definition of weak-tameness which will be used in Chapters 5 and 7:

Definition 3.19. *A source is weakly Λ -tame with width δ if the abscissa of absolute convergence of the Dirichlet series $\Lambda(s)$ is equal to $1 + \delta$.*

A source is weakly Π -tame with width δ if the abscissa of absolute convergence of the Dirichlet series $\Pi(s)$ is equal to δ .

Furthermore, as the two series Λ and Π satisfy the relations

$$|\Pi(s)| \leq \Lambda(\Re s) \quad (\text{for } \Re s \geq 0), \quad \text{and} \quad |\Lambda(s)| \leq \Pi(\Re s - 1) \quad (\text{for } \Re s \geq 1), \quad (3.11)$$

this entails the following inequality between the two widths $\delta(\Pi)$ and $\delta(\Lambda)$, namely

$$\delta(\Pi) \geq \delta(\Lambda).$$

These definitions are in fact very natural, since they describe various possible behaviors of classical sources. “Most of the time”, the simple sources (memoryless sources or aperiodic Markov chains) are both weakly Λ -tame and Π -tame with width $\delta = 0$. However, there exist “natural” sources, for example intermittent sources described in Section 2.2.3 which are Π -tame with width $\delta = 1/a$ and weakly Λ -tame with $\delta = 0$ as soon as $a > 1$.

Conclusion of the Chapter. We have described the tameness of sources, and announced it was a key property to apply the Rice method. The next chapter describes all the main analytical tools which will be used in our analyses. In particular the Rice method can be viewed as the central method of the thesis. This explains why tameness which was deeply studied in this Chapter is also a key tool in our work.

Chapter 4

Presentation of the main tools

Contents

4.1	Poisson and Bernoulli models	68
4.1.1	Definition of the models.	68
4.1.2	Poisson model.	68
4.1.3	Generating functions.	70
4.1.4	Return from the Poisson model to the Bernoulli model.	70
4.2	Depoissonization.	71
4.2.1	A general depoissonization result.	71
4.2.2	The \mathcal{JS} conditions.	71
4.2.3	Analytic extension of the sequence $T(n)$	72
4.3	Binomial recurrence and Rice method	72
4.3.1	Binomial recurrence.	72
4.3.2	First step: An integral form.	72
4.3.3	Step 2. Shifting to the left.	74
4.3.4	Tameness of $\varpi(s)$	74
4.3.5	Asymptotic estimates for $T(n)$	75
4.4	Proof of the asymptotic estimates in the Rice method.	76
4.4.1	The statement	76
4.4.2	Estimates near the real axis.	77
4.4.3	Far from the real axis.	77
4.4.4	Proof of Proposition 4.9.	78
4.5	Mellin transform	79
4.5.1	Definition of the Mellin transform	79
4.5.2	Main properties of the Mellin transform.	80
4.5.3	Inverse Mellin transform	81
4.5.4	Asymptotic expansion of f with its Mellin transform f^*	81
4.6	The “Poisson–Mellin–Newton–Rice” cycle.	82
4.6.1	The Valuation-Degree Condition	82
4.6.2	Existence and expression of the analytic lifting $\varpi(s)$	82

In this chapter, we first define the probabilistic models for our analyses. The initial model is the Bernoulli model where the number of keys (now words) is fixed. As we already mentioned, the Poisson model where the number of words follows a Poisson law is usually easier to deal with. Very often, one performs the main analyses in the Poisson model, and then we wish to return to the (natural) Bernoulli model.

The following of the chapter is then devoted to this return from the Poisson model to the Bernoulli model. We first present de-poissonization methods in Section 4.2, then we focus on the method which will be mainly used in this thesis: the Rice method, which is described in Section 4.3, and proven in Section 4.4. The Mellin transform was a tool which underlies the two technics – Depoissonization technics, as well as Rice method –. This is why we recall the main facts on this transform in Section 4.5. We conclude by a description of the “Poisson-Mellin-Newton-Rice” cycle which gathers all the main tools of this chapter and is central in our work.

4.1 Poisson and Bernoulli models

4.1.1 Definition of the models.

When the cardinal n of \mathcal{V} is fixed, and words $V_i \in \mathcal{V}$ are independently emitted by the source \mathcal{S} , this is the Bernoulli model denoted by $(\mathcal{B}_n, \mathcal{S})$. However, it proves technically convenient to consider that the sequence \mathcal{V} has a variable number N of elements that obeys a Poisson law of rate Z , which is defined as

$$\mathbb{P}[N = k] = e^{-Z} \frac{Z^k}{k!}. \quad (4.1)$$

4.1.2 Poisson model.

In this model, called the Poisson model of rate Z , the rate Z plays a role much similar to the cardinal of \mathcal{V} . When it is relative to probabilistic source \mathcal{S} , the model, denoted by $(\mathcal{P}_Z, \mathcal{S})$, is composed with two main steps:

- (a) The number N of words is drawn according to the Poisson law of rate Z ;
- (b) Then, the N words are independently drawn from the source \mathcal{S} .

There is an important property of the Poisson law. We know that each infinite word is associated to a parameter that is a real in the interval $]0, 1[$.

Lemma 4.1. *Consider the Poisson model $(\mathcal{P}_Z, \mathcal{S})$, and, for subintervals \mathcal{J}, \mathcal{K} of \mathcal{I} , of respective lengths u, t , denote by $N_{\mathcal{J}}, N_{\mathcal{K}}$ the number of words of the source whose parameter belongs to the intervals \mathcal{J}, \mathcal{K} . Then:*

- (i) *The random variable $N_{\mathcal{J}}$ follows a Poisson law of parameter Zu .*
- (ii) *For disjoint subintervals \mathcal{I}, \mathcal{J} , with $\mathcal{I} \cap \mathcal{J} = \emptyset$, the random variables $N_{\mathcal{I}}$ and $N_{\mathcal{K}}$ are independent, namely*

$$\mathbb{P}[N_{\mathcal{I}} = k \quad \text{and} \quad N_{\mathcal{J}} = \ell] = \mathbb{P}[N_{\mathcal{I}} = k] \cdot \mathbb{P}[N_{\mathcal{J}} = \ell].$$

Proof. We first recall the binomial and the multinomial laws which hold in the Bernoulli model $(\mathcal{B}_n, \mathcal{S})$,

$$\mathbb{P}_{[n]}[N_{\mathcal{J}} = k] = \binom{n}{k} u^k (1-u)^{n-k}, \quad \mathbb{P}_{[n]}[N_{\mathcal{I}} = k \text{ and } N_{\mathcal{J}} = \ell] = \binom{n}{k, \ell, n-k-\ell} u^k t^\ell (1-u-t)^{n-k-\ell}.$$

Assertion (i). We begin with conditional probabilities,

$$\begin{aligned} \mathbb{P}_Z[N_{\mathcal{J}} = k] &= \sum_{n \geq k} \mathbb{P}_Z[N_{\mathcal{J}} = k \mid N = n] \mathbb{P}_Z[N = n] \\ &= \sum_{n \geq k} \mathbb{P}_{[n]}[N_{\mathcal{J}} = k] \mathbb{P}_Z[N = n]. \end{aligned}$$

Using the Definition of the Poisson law, and the binomial law of the Bernoulli model,

$$\mathbb{P}_Z[N_{\mathcal{J}} = k] = \sum_{n \geq k} e^{-Z} \frac{Z^n}{n!} \binom{n}{k} u^k (1-u)^{n-k}.$$

Then we distribute the powers of Z^n and the factors of the binomial coefficient

$$\mathbb{P}_Z[N_{\mathcal{J}} = k] = e^{-Z} \frac{(Zu)^k}{k!} \sum_{n \geq k} \frac{Z^{n-k}}{(n-k)!} (1-u)^{n-k}.$$

We recover the expression of $e^{Z(1-u)}$, and finally

$$\begin{aligned} \mathbb{P}_Z[N_{\mathcal{J}} = k] &= \frac{(Zu)^k}{k!} \cdot e^{-Z} \cdot e^{Z(1-u)} \\ &= e^{-Zu} \frac{(Zu)^k}{k!} \end{aligned}$$

This proves *Assertion (i)*.

Assertion (ii). We begin with conditional probabilities,

$$\begin{aligned} \mathbb{P}_Z[N_{\mathcal{J}} = k \text{ and } N_{\mathcal{K}} = \ell] &= \sum_{n \geq k+\ell} \mathbb{P}_Z[N_{\mathcal{J}} = k \text{ and } N_{\mathcal{K}} = \ell \mid N = n] \mathbb{P}_Z[N = n] \\ &= \sum_{n \geq k+\ell} \mathbb{P}_{[n]}[N_{\mathcal{I}} = k \text{ and } N_{\mathcal{J}} = \ell] \mathbb{P}_Z[N = n]. \end{aligned}$$

Using the definition of the Poisson law, and the multinomial law of the Bernoulli model

$$\mathbb{P}_Z[N_{\mathcal{J}} = k \text{ and } N_{\mathcal{K}} = \ell] = e^{-Z} \sum_{n \geq k+\ell} \frac{Z^n}{n!} \binom{n}{k, \ell, n-k-\ell} u^k t^\ell (1-u-t)^{n-k-\ell}.$$

Then we distribute the powers of Z^n and the factors of the multinomial coefficient

$$\mathbb{P}_Z[N_{\mathcal{J}} = k \text{ and } N_{\mathcal{K}} = \ell] = e^{-Z} \frac{(Zu)^k}{k!} \frac{(Zt)^\ell}{\ell!} \sum_{n \geq k+\ell} \frac{Z^{n-k-\ell}}{(n-k-\ell)!} (1-u-t)^{n-k-\ell}.$$

We recover the expression of $e^{Z(1-u-t)}$, and finally

$$\mathbb{P}_Z[N_{\mathcal{J}} = k \text{ and } N_{\mathcal{K}} = \ell] = \frac{(Zu)^k}{k!} \frac{(Zt)^\ell}{\ell!} \cdot e^{-Z} \cdot e^{Z(1-u-t)} = e^{-Zu} \frac{(Zu)^k}{k!} \cdot e^{-Zt} \frac{(Zt)^\ell}{\ell!}.$$

This proves *Assertion (ii)*. □

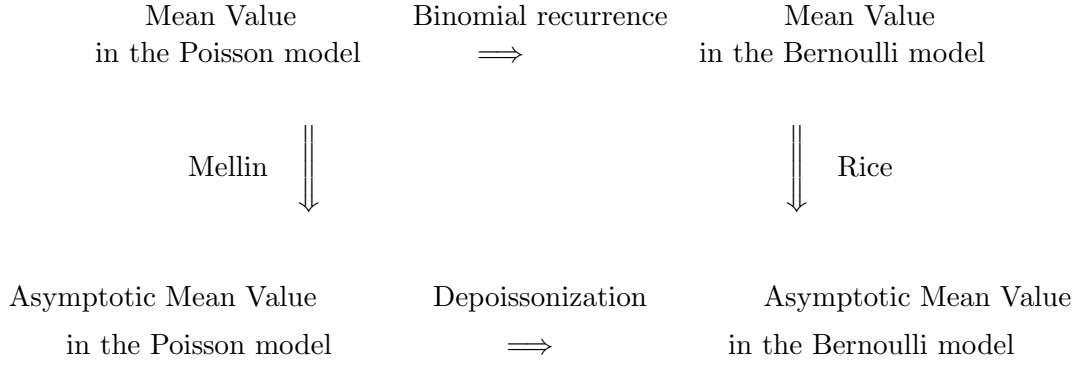


Figure 4.1: Possible ways to obtain the asymptotic mean value in the Bernoulli model from the exact mean value in the Poisson model.

4.1.3 Generating functions.

Consider a sequence $T(n)$ and denote by $B(z), P(z)$ the two following generating functions

$$B(z) := \sum_{n \geq 0} T(n) \frac{z^n}{n!}, \quad P(z) := e^{-z} B(z) = e^{-z} \sum_{n \geq 0} T(n) \frac{z^n}{n!}.$$

The first one is the exponential generating function of the sequence $T(n)$ whereas the second one is the Poisson generating function of the sequence $T(n)$.

If now $T(n)$ is the expectation of some random variable R in the Bernoulli model $(\mathcal{B}_n, \mathcal{S})$, then $P(Z)$ is the expectation of the same random variable in the Poisson model $(\mathcal{P}_Z, \mathcal{S})$ of rate Z . Indeed, with conditional probabilities

$$\mathbb{E}_Z[R] = \sum_{n \geq 0} \mathbb{E}_Z[R \mid N = n] \mathbb{P}_Z[N = n] = e^{-Z} \sum_{n \geq 0} \mathbb{E}_{[n]}[R] \frac{Z^n}{n!} = e^{-Z} \sum_{n \geq 0} T(n) \frac{Z^n}{n!} = P(Z).$$

4.1.4 Return from the Poisson model to the Bernoulli model.

Assume now that the expectation of the random variable R is known in the Poisson model. Is it easy to return to the Bernoulli model $(\mathcal{B}_n, \mathcal{S})$ and obtain the expectation of R in the Bernoulli model? There are two possibilities

- (a) The expression of the series $P(Z)$ is known, via its series expansion and its coefficients $\varphi(n)$, defined as

$$\varphi(n) := (-1)^n [Z^n] P(Z), \quad \text{with the expansion} \quad P(Z) := \sum_{n \geq 0} (-1)^n \varphi(n) \frac{Z^n}{n!}.$$

Given these coefficients, an exact expression in the Bernoulli model is obtained through the use of a binomial recurrence (see Section 4.3.1).

- (b) The asymptotics of the series $P(Z)$ is known as $Z \rightarrow \infty$, in particular in *cones*. Then, it is possible to depoissonize and prove, under some natural conditions, that the two sequences $T(n)$ and $P(n)$ behave in the same asymptotic way.

The diagram of Figure 4.1 is very useful for understanding the two points of view.

4.2 Depoissonization.

We first focus in the path (b) and we consider depoissonization techniques. This method is used when one has a good knowledge of the Poisson transform $P(z)$ itself (whereas the Rice method will be used when one has a good knowledge of its coefficients $\varphi(n)$). As previously, we wish to return to the Bernoulli model, and we can expect that $P(n)$ which is the expectation of the Poisson model (when the cardinality N follows a Poisson law of rate $z = n$) is close to $T(n)$ (which is the expectation in the Bernoulli model when N is fixed and equal to n).

This return to the Bernoulli model needs a good behaviour of $P(z)$ with respect to cones. For $\theta < \pi/2$, the cone S_θ is the set of complex numbers z whose argument $\arg z$ satisfies the inequality $|\arg z| \leq \theta$.

4.2.1 A general depoissonization result.

We use the following theorem of Jacquet and Szpankowski. This is a new result [44] which is very helpful; and greatly simplifies the previous depoissonization results due to the same authors, described in [45] and [46].

Theorem 4.2 (Jacquet and Szpankowski, 2014). *Let $P(z)$ be the Poisson transform of a sequence $T(n)$, namely*

$$P(z) = e^{-z} \sum_{n \geq 0} T(n) \frac{z^n}{n!}.$$

This series is assumed to be absolutely convergent for $z \in \mathbb{C}$. Assume moreover that the sequence $T(n)$ admits an analytic extension $z \mapsto T(z)$ in a cone S_α , which is of polynomial growth in this cone S_α when $|z| \rightarrow \infty$, i.e., there exists γ such that, for $|z| \rightarrow \infty$

$$|\arg(z)| \leq \alpha \implies |T(z)| = O(|z|^\gamma).$$

Then, there exists a cone S_β where the following estimate holds for $|z| \rightarrow \infty$,

$$P(z) = T(z) + \frac{z}{2} T''(z) + O(z^{\gamma-2}).$$

4.2.2 The \mathcal{JS} conditions.

This theorem is based itself on two previous results of the same authors. The first one describes two sufficient conditions (a) and (b) sometimes called the \mathcal{JS} conditions under which one can relate the asymptotic behaviours of P and T .

Theorem 4.3 (Jacquet and Szpankowski, 1998). *Let $P(z)$ be the Poisson transform of a sequence $T(n)$, namely*

$$P(z) = e^{-z} B(z), \quad \text{with} \quad B(z) = \sum_{n \geq 0} T(n) \frac{z^n}{n!},$$

that is assumed to be an entire function of z . It is also assumed that there exists $\theta \in]0, \pi/2[$ for which the two following conditions simultaneously hold in a linear cone S_θ , for some real numbers $a, b, r > 0, \beta$, and $\delta < 1$:

- (i) For $z \in S_\theta$, $|z| > r \implies |P(z)| \leq b |z|^\beta$.
- (ii) For $z \notin S_\theta$, $|z| > r \implies |P(z)e^z| = |B(z)| \leq a \exp(\delta |z|)$.

Then, one has $T(n) \sim P(n)$ for $n \rightarrow \infty$.

4.2.3 Analytic extension of the sequence $T(n)$.

We now explain how it is possible to obtain sufficient conditions under which Conditions (i) and (ii) are fulfilled.

Condition (i) is proven to hold with the use of Mellin transform. We study the asymptotics of $P(n)$ (for $n \rightarrow \infty$) with the Mellin transform P^* of the function $z \mapsto P(z)$.

Condition (ii) is proven to hold as soon as the coefficients $T(n)$ satisfy the following: There exists a function $z \mapsto T(z)$ which exists in a linear cone, is analytic there and is of polynomial growth when $|z| \rightarrow \infty$. More precisely:

Theorem 4.4 (Jacquet and Spzankowski, 1999). *Let $T(z)$ be an analytic continuation of a sequence $T(n)$ which is $O(|z|^\gamma)$ in a linear cone. Then, for some θ_0 , and for all linear cones S_θ with $\theta < \theta_0$, there exist $\delta < 1$ and $a > 0$ such that the exponential generating function $B(z)$ of $T(n)$ satisfies*

$$z \notin S_\theta \implies |B(z)| \leq a \exp(\delta|z|).$$

4.3 Binomial recurrence and Rice method

4.3.1 Binomial recurrence.

We now focus on the Case (a) described in Section 4.1.4. In this case, the expression of the series $P(z)$ is known, via its coefficients $\varphi(n)$, as

$$\varphi(n) := (-1)^n [z^n]P(z), \quad P(z) := \sum_{n \geq 0} (-1)^n \varphi(n) \frac{z^n}{n!}.$$

It is then easy to recover the coefficients $T(n)$ (and thus the expectation of R in the Bernoulli model) with the equality between the two series

$$P(z) = e^{-z} B(z) \quad \text{and then} \quad B(z) = e^z P(z).$$

This entails the following relations between the coefficients $T(n)$ and $\varphi(n)$

$$\varphi(n) = \sum_{k \leq n} \binom{n}{k} (-1)^k T(k), \quad T(n) = \sum_{k \leq n} \binom{n}{k} (-1)^k \varphi(k).$$

This gives rise to a binomial recurrence.

4.3.2 First step: An integral form.

For a sequence T of general term $T(k)$, with $T(k) \in \mathbb{C}$, the valuation $\text{val}(T)$ is the smallest index of non-zero elements of T . We assume in the following that $\text{val}(T)$ is at least equal to 2. This is due to the fact that the cost of sorting or searching algorithms on a subsequence of cardinality strictly less than 2 equals zero. We then let in the following $1 + \sigma_0 := \text{val}(T)$, with an integer $\sigma_0 \geq 1$. The first step of the Rice method transforms the sum of the binomial recurrence

$$T(n) = \sum_{k=1+\sigma_0}^n (-1)^k \binom{n}{k} \varphi(k).$$

into an integral form [53, 54]. When the sequence $\varphi(k)$ admits an analytic lifting, this is just an (easy) application of the Residue formula.

Proposition 4.5. *Let $T(n)$ be a numerical sequence which can be written as*

$$T(n) = \sum_{k=1+\sigma_0}^n \binom{n}{k} (-1)^k \varphi(k).$$

Assume that the sequence $(\varphi(k))$ admits an analytic lifting $\varpi(s)$ in the half plane $\Re s > \sigma_1$ with $\sigma_0 < \sigma_1 < 1 + \sigma_0$, which is there of polynomial growth with order at most r . Then the sequence $T(n)$ admits a Nörlund-Rice representation, for $n > r + 1$ and any d with $d \in]\sigma_1, 1 + \sigma_0[$

$$T(n) = -\frac{1}{2i\pi} \int_{d-i\infty}^{d+i\infty} L_n(s) \cdot \varpi(s) ds \quad \text{with} \quad L_n(s) := \frac{(-1)^n n!}{s(s-1) \dots (s-n)} \quad (4.2)$$

The function $L_n(s)$ is called the *Rice Kernel*.

Proof. The proof has two main steps. We begin with the following lemma which transforms the sum $T(n)$ into an integral over a rectangle \mathcal{C} which encircles the segment $[1 + \sigma_0, n]$.

Lemma 4.6. *Let $T(n)$ be a numerical sequence which can be written as*

$$T(n) = \sum_{k=1+\sigma_0}^n \binom{n}{k} (-1)^k \varphi(k).$$

Assume that the sequence $(\varphi(k))$ admits an analytic lifting $\varpi(s)$ in the half plane $\Re s > \sigma_1$ with $\sigma_0 < \sigma_1 < 1 + \sigma_0$. Then $T(n)$ admits the integral representation

$$T(n) = \frac{1}{2i\pi} \int_{\rho_M} L_n(s) \cdot \varpi(s) ds \quad (4.3)$$

where $L_n(s)$ is defined in (4.2) and ρ_M is a positively oriented rectangle defined by the two vertical lines $\Re s = d$ (with $d \in]\sigma_1, 1 + \sigma_0[$), $\Re s = n + M$ and two horizontal lines $\Im s = \pm M$

Proof. Consider the rectangle ρ_M and denote by \mathcal{C}_M its interior. Since $\varpi(s)$ is analytic on the half plane $\Re s > \sigma_1$, the set of poles of the integrand

$$L_n(s) \cdot \varpi(s) := \frac{(-1)^n n! \varpi(s)}{s(s-1) \dots (s-n)}$$

inside the rectangle \mathcal{C}_M is the set $\{1 + \sigma_0, \dots, n\}$. The sum of residues of the integrand inside the rectangle \mathcal{C}_M is

$$\sum_{k=1+\sigma_0}^n \text{Res}(L_n(s) \cdot \varpi(s); s = k).$$

However, for an integer $k \in \{1 + \sigma_0, \dots, n\}$, the equality holds

$$\text{Res}(L_n(s) \cdot \varpi(s); s = k) = (-1)^n (-1)^{n-k} \frac{n! \varpi(k)}{k! (n-k)!} = (-1)^k \binom{n}{k} \varpi(k).$$

Then, the Residue Theorem proves the Lemma. \square

The second step for the proof of Proposition 4.5 transforms the rectangle ρ_M into a vertical line. When $M \rightarrow \infty$, the horizontal lines are pushed to $\pm\infty$, then the rightmost boundary to $+\infty$. Using the fact that $\varpi(s)$ is of polynomial growth, the integral of $L_n(s) \cdot \varpi(s)$ on the positively oriented contour ρ_M tends to the integral of $L_n(s) \cdot \varpi(s)$ on the vertical line $\Re s = d$ oriented from the north to the south. This ends the proof of Proposition 4.5. \square

4.3.3 Step 2. Shifting to the left.

The previous proposition shows how to transform a binomial sum into an integral in the complex plane. For any real $d \in]\sigma_1, 1 + \sigma_0[$, one has

$$T(n) := \sum_{k=1+\sigma_0}^n (-1)^k \binom{n}{k} \varphi(k) = -\frac{1}{2i\pi} \int_{\Re s=d} L_n(s) \cdot \varpi(s) ds, \quad (4.4)$$

where $L_n(s)$ is the Rice kernel defined in (4.2) and $\varpi(s)$ is the analytic lifting of the sequence $\varphi(k)$. Then, along general principles in analytic combinatorics [27, 29], the integration line can be pushed to the left, as soon as $L_n(s) \cdot \varpi(s)$ (closely related to $\varpi(s)$) has good analytic properties: we need a region \mathcal{R} on the left of $\Re s = \sigma_0$, where $\varpi(s)$ is of polynomial growth (for $|\Im s| \rightarrow \infty$) and meromorphic. With a good knowledge of its poles, we finally obtain a residue formula

$$T(n) = - \left[\sum_k \operatorname{Res} [L_n(s) \cdot \varpi(s) : s_k] + \frac{1}{2i\pi} \int_{\mathcal{C}_2} L_n(s) \cdot \varpi(s) ds \right], \quad (4.5)$$

where \mathcal{C}_2 is a curve of class \mathcal{C}^1 enclosed in \mathcal{R} oriented from the south to the north and the sum is extended to all poles s_k of $L_n(s) \cdot \varpi(s)$ inside the domain delimited by the vertical line $\Re s = \sigma_1$ and the curve \mathcal{C}_2 .

The dominant singularities of $L_n(s) \cdot \varpi(s)$ provide the asymptotic behaviour of $T(n)$, and the remainder integral is estimated using the polynomial growth of $L_n(s) \cdot \varpi(s)$ when $|\Im(s)| \rightarrow \infty$.

4.3.4 Tameness of $\varpi(s)$.

To shift the integration path to the left, we need a region \mathcal{R} on the left of $\Re s = \sigma_0$, where $\varpi(s)$ is of polynomial growth (for $|\Im s| \rightarrow \infty$) and meromorphic. We need also a good knowledge of its poles. These properties are described by the tameness of the function $\varpi(s)$ (illustrated in Figure 4.2).

Definition 4.7. A function $\varpi(s)$ is tame at σ_0 if one of the three following properties holds:

(a) [S-shape] (shorthand for Strip shape) there exists a vertical strip $\Re(s) > \sigma_0 - \delta$ for some $\delta > 0$ where $\varpi(s)$ is meromorphic, has a sole possible pole (of order $k_0 \geq 0$) at $s = \sigma_0$ and is of polynomial growth as $|\Im s| \rightarrow +\infty$.

(b) [H-shape] (shorthand for Hyperbolic shape) there exists an hyperbolic region \mathcal{R} , defined as, for some $A, B, \rho > 0$

$$\mathcal{R} := \{s = \sigma + it; \quad |t| \geq B, \quad \sigma > \sigma_0 - \frac{A}{t^\rho}\} \cup \{s = \sigma + it; \quad \sigma > \sigma_0 - \frac{A}{B^\rho}, |t| \leq B\},$$

where $\varpi(s)$ is meromorphic, with a sole possible pole (of order $k_0 \geq 0$) at $s = \sigma_0$ and is of polynomial growth in \mathcal{R} as $|\Im s| \rightarrow +\infty$.

(c) [P-shape] (shorthand for Periodic shape) there exists a vertical strip $\Re(s) > \sigma_0 - \delta$ for some $\delta > 0$ where $\varpi(s)$ is meromorphic, has only a pole (of order $k_0 \geq 0$) at $s = \sigma_0$ and a family (s_k) (for $k \in \mathbb{Z} \setminus \{0\}$) of simple poles at points $s_k = \sigma_0 + 2ki\pi t$ with $t \neq 0$, and is of polynomial growth as $|\Im s| \rightarrow +\infty$ ¹.

There are four parameters relative to the tameness: the real σ_0 is the position, the integer k_0 is the order, and, when they exist, the real δ is the width, and the real ρ is the exponent.

¹More precisely, this means that $\varpi(s)$ is of polynomial growth on a family of horizontal lines $t = t_k$ with $t_k \rightarrow \infty$, and on vertical lines $\Re(s) = \sigma_0 - \delta'$ with some $\delta' < \delta$.

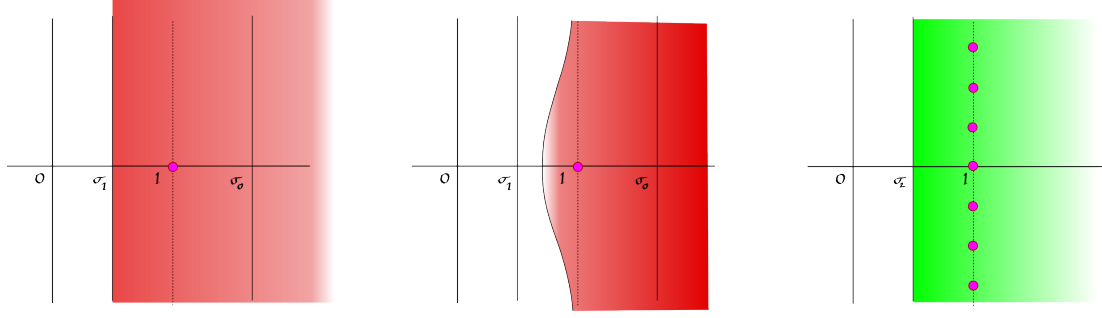


Figure 4.2: Three possible domains where the function $\varpi(s)$ is analytic and of polynomial growth.

4.3.5 Asymptotic estimates for $T(n)$.

The Rice kernel defined in (4.2) plays a center role in Section 4.4. The position σ_0 of tameness of $\varpi(s)$ will also be an integer $\sigma_0 \in \{1, 2\}$. In this case, the tameness of $\varpi(s)$ entails the tameness of $L_n(s) \varpi(s)$. The tameness parameters for $L_n(s) \cdot \varpi(s)$ are closely related to those of $\varpi(s)$: the position, the width and the exponent are the same. We also remark that the tameness order for $L_n(s) \cdot \varpi(s)$ equals $k_0 + 1$, where k_0 is the tameness order of $\varpi(s)$. With the tameness properties of $L_n(s) \cdot \varpi(s)$, the shifting to the left inside the tameness region is possible and leads to the result stated in Proposition 4.8. Note that, from Eq. (4.5), the dominant part of the asymptotics comes from considering poles in the region \mathcal{R} (residue calculus) and the error term comes from the evaluation of the integral on the curve \mathcal{C}_2 in Eq. (4.5).

Proposition 4.8. *The following holds for the sequence $T(n)$, when it is related to $\varpi(s)$ by the Rice formula (4.4), with $\sigma_0 \in \{1, 2\}$. If $\varpi(s)$ is tame at $s = \sigma_0$ with order k_0 , then there exists a polynomial Q of degree k_0 such the following asymptotics hold, depending on the tameness shape:*

- (a) *With a S-shape and width δ_0 , for any $\delta < \delta_0$, one has, for $n \rightarrow \infty$,*

$$(-1)^{n+1}T(n) = n^{\sigma_0}Q(\log n) + O(n^{\sigma_0-\delta}).$$

- (b) *With a H-shape and exponent β_0 , then, for any β with $\beta < 1/(\beta_0 + 1)$, one has, for $n \rightarrow \infty$,*

$$(-1)^{n+1}T(n) = n^{\sigma_0}Q(\log n) + O\left(n^{\sigma_0} \cdot \exp[-(\log n)^\beta]\right).$$

- (c) *With a P-shape and width δ_0 , then, for any $\delta < \delta_0$, one has, for $n \rightarrow \infty$,*

$$(-1)^{n+1}T(n) = n^{\sigma_0} (Q(\log n) + \Phi(n)) + O(n^{\sigma_0-\delta}),$$

where $n^{\sigma_0} \cdot \Phi(n)$ is the part of the expansion brought by the family of the non real poles of $G(s)$ located on the vertical line $\Re s = \sigma_0$.

Proof. If $\varpi(s)$ is of polynomial growth in a region \mathcal{R} , the line of integration $\Re s = d$ can be moved to the left until a curve ρ , which lies inside the region \mathcal{R} , provided residues of poles inside \mathcal{R} be taken into account. If $\varpi(s)$ has a pole of order k_0 at $s = \sigma_0$, then $\varpi(s)/(s - \sigma_0)$ has a pole of order $k_0 + 1$, it admits near this pole the singular expression

$$\frac{\varpi(s)}{s - \sigma_0} = \sum_{k=1}^{k_0+1} \frac{a_k}{(s - \sigma_0)^k},$$

and this pole contributes with a quantity of the form

$$n^{\sigma_0} \left[\sum_{k=0}^{k_0} (-1)^k \frac{a_k}{k!} \log^k n \right].$$

In the cases (a) or (c), the curve ρ can be chosen as a vertical line of equation $\Re s = \alpha$ with $\alpha = \sigma_0 - \delta$. In the case (b), the curve ρ can be chosen as an hyperbolic curve of the form

$$\rho := \left\{ s = \sigma + it, |t| \geq B, \sigma = \sigma_0 - \frac{A}{|t|^{\beta_0}} \right\} \cup \left\{ s = \sigma + it, \sigma = \sigma_0 - \frac{A}{B^{\beta_0}}, |t| \leq B \right\},$$

for some strictly positive constants (A, B, β_0) . The remainder of the proof is devoted to the computation of the integral

$$\int_{\rho} L_n(s) \cdot \varpi(s) ds.$$

Proposition 4.9 proves the following : if $\varpi(s)$ is of polynomial growth on the curve ρ as $|s| \rightarrow \infty$, this integral is of order $n^{\sigma_0} O(n^{-\delta})$ in the cases (a) and (c) of Proposition 4.8. It is of order $n^{\sigma_0} O(\exp[-(\log n)^{\beta}])$ with $\beta < 1/(1 + \beta_0)$ in the case (b). \square

4.4 Proof of the asymptotic estimates in the Rice method.

4.4.1 The statement

Proposition 4.9. *Associate with a function $\varpi(s)$ the function $L_n(s) \varpi(s)$, where $L_n(s)$ is the Rice kernel defined in (4.2).*

- (i) *Consider the vertical line $\Re s = \alpha$, and assume that $\varpi(s)$ be continuous on $\Re s = \alpha$ and be of at most polynomial growth there: $\varpi(s) = O(s^r)$ as $|s| \rightarrow \infty$ on $\Re s = \alpha$. Then the integral on the vertical $\Re s = \alpha$ of $L_n(s) \cdot \varpi(s)$ admits the following estimate, as $n \rightarrow \infty$:*

$$\int_{\Re s = \alpha} L_n(s) \cdot \varpi(s) ds = O(n^{\alpha}).$$

- (ii) *Consider a curve ρ of hyperbolic type, namely of the form:*

$$\rho := \left\{ s = \sigma + it, |t| \geq B, \sigma = \sigma_0 - \frac{A}{|t|^{\beta_0}} \right\} \cup \left\{ s = \sigma + it, \sigma = \sigma_0 - \frac{A}{B^{\beta_0}}, |t| \leq B \right\},$$

for some strictly positive constants (A, B, β_0) , and assume that $\varpi(s)$ be continuous on ρ and be of (at most) polynomial growth there $\varpi(s) = O(|s|^r)$ as $|s| \rightarrow \infty$. Then the integral of $L_n(s) \cdot \varpi(s)$ on the curve ρ admits the following estimate, as $n \rightarrow \infty$,

$$\int_{\rho} L_n(s) \cdot \varpi(s) ds = n^{\sigma_0} \cdot O(\exp[-(\log n)^{\beta}]), \quad \text{with } \beta < \frac{1}{1 + \beta_0}.$$

For the proof, we only need to consider the upper half-plane. We use $T = \sqrt{n}$ as a cut-off point and decompose each of the curves –the vertical line or the hyperbolic curve ρ – into two parts. The first result provides estimates when s is near the real axis ($|\Im s| \leq T$) and the second lemma deals with the case when s is far from the real axis ($|\Im s| \geq T$) [72].

4.4.2 Estimates near the real axis.

Lemma 4.10. *For s outside of a fixed sector containing the negative real axis in its interior, and under the condition $|s| \leq \sqrt{n}$, one has, as $n \rightarrow \infty$:*

$$L_n(s) := \frac{n!(-1)^n}{s(s-1)\dots(s-n)} = -n^s \Gamma(-s) \left(1 + O\left(\frac{1}{\sqrt{n}}\right) + O\left(\frac{s^2}{n}\right) \right). \quad (4.6)$$

Also, for any s fixed with $s \notin \mathbb{N}$, one has

$$L_n(s) = -n^s \Gamma(-s) \left(1 + O\left(\frac{1}{n}\right) \right). \quad (4.7)$$

Proof. One has

$$L_n(s) = \frac{(-1)^n n!}{s(s-1)\dots(s-n)} = -\frac{n!}{-s(-s+1)\dots(-s+n)} = -\frac{\Gamma(n+1)\Gamma(-s)}{\Gamma(n-s+1)}.$$

Stirling's formula holds in the complex plane, provided a sector around the negative real axis is avoided. Under this condition, one has

$$\Gamma(w+1) = w^w e^{-w} \sqrt{2\pi w} \left(1 + O\left(\frac{1}{n}\right) \right), \quad |w| \rightarrow +\infty. \quad (4.8)$$

With the Stirling formula,

$$\begin{aligned} \frac{\Gamma(n+1)}{\Gamma(n-s+1)} &= \frac{n^n e^{-n} \sqrt{2\pi n}}{(n-s)^{n-s} e^{s-n} \sqrt{2\pi(n-s)}} \left(1 + O\left(\frac{1}{n}\right) \right) \\ &= \exp[n \log n - (n-s) \log(n-s) - s] \left(1 + O\left(\frac{1}{\sqrt{n}}\right) \right) \\ &= \exp[s \log n - (n-s) \log(1+s/n) - s] \left(1 + O\left(\frac{1}{\sqrt{n}}\right) \right). \end{aligned}$$

In the region under consideration, we have $s/n = O(1/\sqrt{n})$, which is a small quantity, so that $\log(1+s/n) = s/n + O(s^2/n^2)$. Consequently,

$$\begin{aligned} \frac{\Gamma(n+1)}{\Gamma(n-s+1)} &= n^s \exp \left[O\left(\frac{s^2}{n}\right) \right] \left(1 + O\left(\frac{1}{\sqrt{n}}\right) \right) \\ &= n^s \left(1 + O\left(\frac{1}{\sqrt{n}}\right) + O\left(\frac{s^2}{n}\right) \right), \end{aligned}$$

and the estimate (4.6) results. The proof of (4.7) is similar, even simpler, via the relation $s/n = O(1/n)$. \square

4.4.3 Far from the real axis.

Lemma 4.11. *Fix any number $m > 0$. Then, there exists a computable constant $K_m > 0$ such that for n large enough, $s = b + it$, b fixed and $t \geq \sqrt{n}$, one has*

$$|L_n(s)| \leq \frac{K_m}{t^m} e^{-L\sqrt{n}}, \quad \text{for } L_n(s) = (-1)^n \frac{n!}{s(s-1)(s-2)\dots(s-n)},$$

with $L = \log(\sqrt{2})$.

Proof. Choose an integer $m > 0$ and set $A = \lfloor \sqrt{n} \rfloor$. We write

$$|L_n(s)| = \frac{1}{|s|} \prod_{a=1}^m \left| \frac{a}{a-s} \right| \prod_{a=m+1}^{m+A} \left| \frac{a}{a-s} \right| \prod_{a=m+A+1}^n \left| \frac{a}{a-s} \right|.$$

The first product has a trivial bound

$$\prod_{a=1}^m \left| \frac{a}{a-s} \right| < \frac{m!}{t^m}. \quad (4.9)$$

For the second product, the complex s is close to the imaginary axis when $n \rightarrow \infty$. The triangle $(a, 0, s)$ is approximately a right triangle. The angle β at a satisfies, for n large enough,

$$\tan(\beta) \sim \frac{|s|}{|a|} \geq 1, \quad \text{and thus} \quad \left| \frac{a}{a-s} \right| = \cos(\beta) < \cos\left(\frac{\pi}{4}\right) = \left(\frac{1}{\sqrt{2}}\right)^A.$$

resulting in

$$\prod_{a=m+1}^{m+A} \left| \frac{a}{a-s} \right| < \left(\frac{1}{\sqrt{2}}\right)^A. \quad (4.10)$$

For the third product, we plainly use the triangle inequality, which gives $|a/(a-s)| < 1$ and

$$\prod_{a=m+A+1}^n \left| \frac{a}{a-s} \right| < 1. \quad (4.11)$$

Collecting (4.9), (4.10), (4.11), we have:

$$|L_n(s)| < \frac{m!}{t^m} \left(\frac{1}{\sqrt{2}}\right)^A = \frac{m!}{t^m} e^{-L\sqrt{n}}.$$

Then, $K_m = m!$ and $L = \log(\sqrt{2})$. □

4.4.4 Proof of Proposition 4.9.

We only need to consider the upper half-plane. We use $T = \sqrt{n}$ as a cut-off point and decompose each positive part $\tilde{\rho}$ of the curve –the vertical line or the hyperbolic curve ρ – into two parts.

Case of a vertical line. We use the decomposition

$$\int_{\tilde{\rho}} L_n(s) \cdot \varpi(s) ds = \int_{s=\alpha}^{\alpha+iT} L_n(s) \cdot \varpi(s) ds + \int_{s=\alpha+iT}^{\alpha+i\infty} L_n(s) \cdot \varpi(s) ds.$$

Near the real axis, namely for $s \in [\alpha, \alpha + iT]$, we apply Lemma 4.10:

$$\int_{s=\alpha}^{\alpha+iT} L_n(s) \cdot \varpi(s) ds = - \int_{s=\alpha}^{\alpha+iT} n^s \Gamma(-s) \varpi(s) (1 + O(n^{-1})) ds \quad (4.12)$$

As the fast decay of $\Gamma(s)$ compensates more for the polynomial growth of $\varpi(s)$ and $|n^s| = n^\alpha$, the integral is $O(n^\alpha)$.

Far from the real axis, namely for $s \in [\alpha + iT, \alpha + i\infty]$, we apply Lemma 4.11:

$$\int_{s=\alpha+iT}^{\alpha+i\infty} |L_n(s) \cdot \varpi(s)| ds < K_m e^{-L\sqrt{n}} \int_{t=T}^{\infty} \frac{t^r}{t^m} dt = O(e^{-L\sqrt{n}}) \quad (4.13)$$

for n large enough, provided m has been chosen such that $m > r + 2$. The combination of Equations (4.12) and (4.13) yields the claimed estimate in the case of a vertical line.

Case of an hyperbolic curve. Consider now the case of an hyperbolic curve, and consider the two parts of the curve $\tilde{\rho}$: the curve ρ^- (near the real axis) and the curve ρ^+ (near imaginary infinity).

$$\int_{\tilde{\rho}} L_n(s) \cdot \varpi(s) ds = \int_{\rho^+} L_n(s) \cdot \varpi(s) ds + \int_{\rho^-} L_n(s) \cdot \varpi(s) ds. \quad (4.14)$$

In the case of the curve ρ^+ , which can be compared to a vertical line, we apply Lemma 4.11 and

$$\left| \int_{\rho^+} L_n(s) \cdot \varpi(s) ds \right| < K_m \int_T^\infty O(t^r) \cdot O(t^{-m}) \cdot e^{-L\sqrt{n}} dt = O\left(e^{-L\sqrt{n}}\right), \quad (4.15)$$

for n large enough, provided m has been chosen such that $m > r + 2$.

Now, near the real axis, Lemma 4.10 gives

$$\int_{\rho^-} L_n(s) \cdot \varpi(s) ds = - \left(\int_{\rho^-} n^s \Gamma(-s) \varpi(s) ds \right) (1 + O(n^{-1})). \quad (4.16)$$

Letting $s := \sigma + it$, and $L := \log n$, we use the following estimates (for some $K > 0$)

$$|n^s| = n^\sigma = n^{\sigma_0} \exp[-ALt^{-\beta_0}], \quad |\varpi(s)\Gamma(-s)| \leq \exp[-Kt].$$

The first one is due to the definition of the curve whereas the second one uses the fast decay of $\Gamma(-s)$ which more than compensates for the polynomial growth of $\varpi(s)$. If we let $L := \log n$, the modulus of the integrand is at most

$$|L_n(s) \cdot \varpi(s)| \leq n^{\sigma_0} \exp[-Kt - ALt^{-\beta_0}].$$

When n (and then L) is fixed, the minimum of the function $t \mapsto Kt + ALt^{-\beta_0}$ is reached for $t^{\beta_0+1} = \beta_0 L/K$. Then the maximum of $|L_n(s) \cdot \varpi(s)|$ is of order $n^{\sigma_0} \exp[-(\log n)^\beta]$ with $\beta < 1/(1 + \beta_0)$. Using the same principles as in the Laplace method (see Sectionchap 8: sec: Laplace method), we obtain the estimate

$$\int_{\rho^-} L_n(s) \cdot \varpi(s) ds = n^{\sigma_0} O(\exp[-(\log n)^\beta]) \quad \text{with } \beta < 1/(1 + \beta_0).$$

this yields the claimed estimate in the case of a hyperbolic curve.

4.5 Mellin transform

The Mellin transform is a central tool in this chapter. We already mentioned it in Section 4.3 about Depoissonization techniques when we said that it is a main tool for obtaining asymptotic estimates of the Poisson transform $P(z)$. Even when we use the Rice method, it is also a central tool since it helps to obtain the analytic lifting of the sequence $\varphi(n)$, as we will explain in Section 4.6.2.

4.5.1 Definition of the Mellin transform

Definition 4.12. Let $f : x \mapsto f(x)$ be a function integrable over $[0, +\infty[$. Then, the Mellin transform of f , denoted f^* , is defined by:

$$\mathcal{M}(f, s) = f^*(s) = \int_0^{+\infty} f(x) x^{s-1} dx.$$

The maximal open vertical strip $\{s : \Re s \in]\alpha, \beta[\}$ where the integral is absolutely convergent is called the fundamental strip. It is also denoted as $\langle \alpha, \beta \rangle$.

When the conditions $f(x) =_{x \rightarrow 0^+} O(x^u)$, $f(x) =_{x \rightarrow +\infty} O(x^v)$ hold, with $u > v$, the Mellin transform f^* exists in the vertical strip $\langle -u, -v \rangle$.

For instance, the Mellin transform of the function $g : x \mapsto e^{-x}$ exists in the strip $\langle 0, +\infty \rangle$ and defines there the Γ function [25][29],

$$\Gamma(s) = \int_0^\infty e^{-x} x^{s-1} dx.$$

4.5.2 Main properties of the Mellin transform.

Lemma 4.13. [Functional properties] *Let f be a function whose Mellin transform f^* admits the fundamental strip $\langle \alpha, \beta \rangle$. Let ρ be a non zero real number, and μ, ν be positive reals. Then the table 4.3 summarizes the main properties of the Mellin transform f^* .*

Function	Mellin transform	Fundamental strip
$f(x)$	$f^*(s)$	$\langle \alpha, \beta \rangle$
$x^\nu f(x)$	$f^*(s + \nu)$	$\langle \alpha - \nu, \beta - \nu \rangle$
$f(x^\rho)$	$\frac{1}{\rho} f^*\left(\frac{s}{\rho}\right)$	$\langle \rho\alpha, \rho\beta \rangle, \rho > 0$
$f\left(\frac{1}{x}\right)$	$f^*(-s)$	$\langle -\beta, -\alpha \rangle$
$f(\mu x)$	$\mu^{-s} f^*(s)$	$\langle \alpha, \beta \rangle, \mu > 0$
$\sum_k \lambda_k f(\mu_k x)$	$\left(\sum_k \lambda_k \mu_k^{-s} \right) f^*(s)$	
$f(x) \log x$	$\frac{d}{ds} f^*(s)$	$\langle \alpha, \beta \rangle$
$\frac{d}{dx} f(x)$	$-(s-1) f^*(s-1)$	$\langle \alpha+1, \beta+1 \rangle$
$\int_0^x f(t) dt$	$-\frac{1}{s} f^*(s+1)$	

Figure 4.3: Properties of the Mellin transform

As an example, consider the function

$$f(x) = \frac{e^{-x}}{1 - e^{-x}} = e^{-x} + e^{-2x} + e^{-3x} + \dots = \sum_{k \geq 1} e^{-kx}.$$

Applying the previous properties, one has, for $\Re(s) > 1$,

$$f^*(s) = \Gamma(s) \left(\frac{1}{1^s} + \frac{1}{2^s} + \frac{1}{3^s} + \dots \right) = \Gamma(s) \left(\sum_{k \geq 1} \frac{1}{k^s} \right) = \Gamma(s) \zeta(s) \quad (s \in \langle 1, +\infty \rangle).$$

Note that the condition $\Re(s) > 1$ entails the absolute convergence of the Mellin transform of f and of the sum defining the zeta function.

4.5.3 Inverse Mellin transform

It is possible to recover f from f^* via an integral along a vertical line $\Re s = c$ which lies inside the fundamental strip of Mellin transform, namely

$$f(x) = \frac{1}{2i\pi} \int_{c-i\infty}^{c+i\infty} f^*(s) x^{-s} ds, \quad \text{when } c \in \langle \alpha, \beta \rangle$$

4.5.4 Asymptotic expansion of f with its Mellin transform f^* .

There is a direct correspondence between the asymptotic expansion of the function f , either at 0 or ∞ and the singularities of its Mellin transform f^* , as we now explain. We begin with the expression of the inverse Mellin transform,

$$f(x) = \frac{1}{2i\pi} \int_{c-i\infty}^{c+i\infty} f^*(s) x^{-s} ds.$$

Proposition 4.14. *Let f be continuous in $]0, +\infty[$ with Mellin transform $f^*(s)$ having a non-empty fundamental strip $\langle \alpha, \beta \rangle$. Assume that $f^*(s)$ admits a meromorphic continuation to $\langle \alpha, \gamma \rangle$ for some $\gamma > \beta$ and is analytic on the vertical line $\Re(s) = \gamma$. Assume also that there exists a real $\eta \in]\alpha, \beta[$ such that*

$$|f^*(s)| = O(|s|^{-r}) \quad \text{with } r > 1 \text{ for } |s| \rightarrow \infty \text{ in the strip } \langle \eta, \gamma \rangle.$$

If $f^*(s)$ admits the singular expression, for $s \in \langle \eta, \gamma \rangle$,

$$f^*(s) = \sum_{(\xi, k) \in \mathcal{A}} d_{\xi, k} \frac{1}{(s - \xi)^k},$$

then an asymptotic expression of $f(x)$ at $+\infty$ is

$$f(x) = \sum_{(\xi, k) \in \mathcal{A}} d_{\xi, k} \left(\frac{(-1)^{k-1}}{(k-1)!} x^{-\xi} (\log x)^k \right) + O(x^{-\gamma}).$$

Singularities of $f^*(s)$	Terms in the asymptotic expansion of $f(x)$
Simple pole at $s = a$	x^{-a}
Pole of order $k + 1$ at $s = a$	$\frac{(-1)^k}{k!} x^{-a} (\log x)^k$
Simple pole at $a = \sigma + it$, with $\sigma, t \in \mathbb{R}$	Fluctuating term $x^{-\sigma} e^{-it \log x}$

For example, when $f(x)$ is defined as

$$f(x) = \sum_{k=1}^{+\infty} (-1)^k \log k e^{-k^2 x},$$

its Mellin transform is

$$f^*(s) = [2^{1-2s}(\log 2)\zeta(2s) + (1 - 2^{1-2s})\zeta'(2s)]\Gamma(s),$$

with

$$\Gamma(s) = \int_0^\infty e^{-x} x^{s-1} dx, \quad \zeta(s) = \sum_{k=1}^\infty \frac{1}{k^s}.$$

This equation is true for $\Re s > 1$, which ensures simultaneously absolute convergence of the Mellin integral and Dirichlet series. As Γ function has simple poles at non positive integers and ζ function has a simple pole at $s = 1$, the integral contour might be legitimately shifted to the left. By sweeping the integration contour till the vertical line $\Re s = \frac{5}{2}$, one takes into account the poles $s = 0, -1, -2$, whose contributions are

$$f(x) = \log \sqrt{\frac{\pi}{2}} + c_1 x + c_2 x^2 + O(x^{\frac{5}{2}}).$$

4.6 The “Poisson–Mellin–Newton–Rice” cycle.

We return here to the Rice method described in 4.3. It needs an analytic lifting for the sequence $\varphi(k)$. We recall that $\varphi(k)$ are the coefficients of the Poisson transform of our initial sequence $T(k)$. As we will see in Chapter 5, the existence of a simple lifting $\varpi(s)$ for the sequence $\varphi(n)$ of the coefficients of the Poisson transform is not always directly granted, but it can be obtained by the “Poisson–Mellin–Newton–Rice” cycle, well described in [28], that we now adapt to our setting.

4.6.1 The Valuation-Degree Condition

This will be an important condition for obtaining such an analytic lifting.

Definition 4.15. For a sequence T of general term $T(k)$, with $T(k) \in \mathbb{C}$,

- (i) the valuation $\text{val}(T)$ is the smallest index of non-zero elements of T .
- (ii) the degree $\text{deg}(T)$ as the infimum of all c such that $T(k) = O(k^c)$.

The Valuation-Degree Condition (VLC) is : the strict equality $\text{val}(T) > \text{deg}(T)$ holds.

This is not a very restrictive hypothesis, since if the inequality $\text{deg}(T) \geq \text{val}(T)$ holds, we can subtract to the sequence $T(k)$ its leading terms so that the new sequence satisfies the inequality $\text{val}(T) > \text{deg}(T)$.

4.6.2 Existence and expression of the analytic lifting $\varpi(s)$

When the inequality $\text{val}(T) > \text{deg}(T)$ holds, we use the Mellin transform $P^*(s)$ of the Poisson generating function $P(z)$ of the sequence $T(k)$ in order to build $\varpi(s)$:

Proposition 4.16. Consider a sequence $T(k)$ which satisfies the inequality $\text{val}(T) > \text{deg}(T)$, and let $\text{val}(T) := 1 + \sigma_0$. Denote by $P(z)$ its Poisson generating function, and by $\varphi(k)$ the coefficients of $P(z)$,

$$P(z) := e^{-z} \sum_{k \geq 1 + \sigma_0} T(k) \frac{z^k}{k!} = \sum_{k \geq 1 + \sigma_0} (-1)^k \varphi(k) \frac{z^k}{k!},$$

so that the binomial relations hold between the two sequences $T(k)$ and $\varphi(k)$, namely

$$T(n) = \sum_{k=1+\sigma_0}^n \binom{n}{k} (-1)^k \varphi(k) \quad \varphi(n) = \sum_{k=1+\sigma_0}^n \binom{n}{k} (-1)^k T(k). \quad (4.17)$$

Then, for any $c \in]\sigma_0, 1 + \sigma_0[$, the sequence $\varphi(k)$ admits an analytic lifting $\varpi(s)$ on the half-plane $\Re s > c$ which involves the Mellin transform $P^*(s)$ under the form

$$\varpi(s) = \frac{P^*(-s)}{\Gamma(-s)}. \quad (4.18)$$

Proof. There are three main steps.

Step 1. We prove that the Mellin transform $P^*(s)$ of the Poisson generating function $P(z)$ exists in the fundamenral strip $\langle -1 - k_0, -c \rangle$ and is well defined by the following formal exchange of integration and summation, which is justified in the fundamntl strip $\langle -1 - k_0, -c \rangle$,

$$P^*(s) = \sum_{k \geq 1+\sigma_0} \frac{T(k)}{k!} \int_0^\infty e^{-z} z^k z^{s-1} dz = \sum_{k \geq 1+\sigma_0} \frac{T(k)}{k!} \Gamma(k+s) \quad (4.19)$$

$$= \Gamma(s) \left[\sum_{k=1+\sigma_0}^\infty T(k) \frac{s(s+1) \dots (s+k-1)}{k!} \right]. \quad (4.20)$$

Indeed, each $\Gamma(k+s)$ is well defined for $k \geq 1 + k_0$ as soon as $\Re s > -1 - k_0$. Furthermore, we use the following estimate, already proven in Lemma 4.10,

$$\frac{s(s+1) \dots (s+n-1)}{n!} = \frac{n^{s-1}}{\Gamma(s)} \left[1 + O\left(\frac{1}{n}\right) \right] \quad (n \rightarrow \infty). \quad (4.21)$$

This estimate holds uniformly with respect to s in a compact subset of $\mathbb{C} \setminus \mathbb{Z}_{\leq 1}$, and this proves that the series of (4.19) is uniform convergent for $\Re s < -c$.

Step 2. If we now let

$$\varpi(s) := \frac{P^*(-s)}{\Gamma(-s)} = \sum_{k=2}^\infty (-1)^k T(k) \frac{s(s-1) \dots (s-k+1)}{k!}, \quad (4.22)$$

the right-side is expressed as a Newton interpolation series converges on the half-plane $\Re s > c$ and defines there an analytic continuation $\varpi(s)$ of the ratio $P^*(-s)/\Gamma(-s)$.

Step 3. With Equations (4.17) and (4.22), the function $\varpi(s)$ satisfies $\varpi(n) = \varphi(n)$ for $n \geq 1 + \sigma_0$. \square

Then, the “Poisson–Mellin–Newton–Rice” cycle provides the analytic lifting $\varpi(s)$ which is used in the Rice method. With the expression given by (4.18) the singularities of $\varpi(s)$ are often apparent. But this “Poisson–Mellin–Newton–Rice” cycle does not prove that $\varpi(s)$ is of polynomial growth. This property is essential for using the Rice method and must be proven by other means.

Conclusion for the first part of the thesis. The first part of the thesis ends here. In these first four chapters, we have described all the tools that will be needed to perform our task: the realistic analysis of sorting and searching algorithms. We recall the plan of the thesis in Figure 4.4.

Before going to the realistic analysis of the studied algorithms **QuickSort**, **QuickSelect**, **BubSort**, **InsSort** and **SelMin**, we describe in Chapter 5 the general method we use for the analysis, with its two steps: in the algebraic combinatoric step, we compute the exact number of symbol comparisons performed by the algorithms, and also the mixed Dirichlet series of the pair “source-algorithm”. Then, we deal with this Dirichlet series, with its poles and its tameness, which makes possible the use of the Rice method. This gives rise to the asymptotics of the mean number of comparisons. Each step of the method is illustrated in the analysis of **SelMin**. Then, we apply the same method in Chapter 6 for sorting algorithms and in Chapter 7 for **QuickSelect** and its variants.

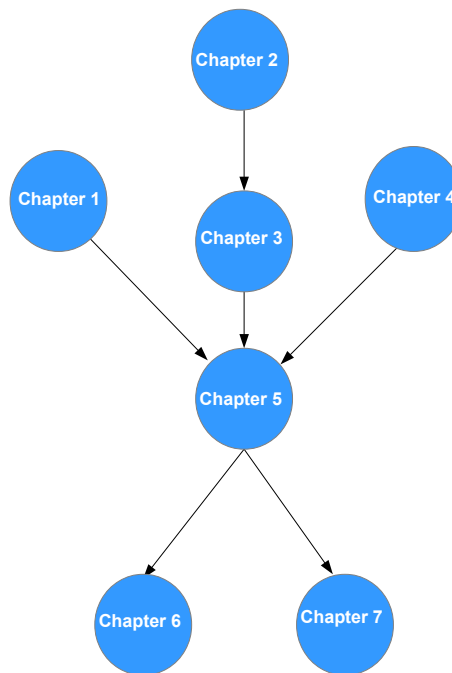


Figure 4.4: Plan of the thesis.

Chapter 5

General framework for a more realistic analysis

Contents

5.1	Main principles.	86
5.1.1	Two probabilistic models for the inputs of the algorithm.	86
5.1.2	The cost of interest.	86
5.1.3	A general scheme for the analysis.	87
5.1.4	Four steps in the analysis.	88
5.1.5	Main objects in the analysis.	88
5.1.6	General Results. The case of the SelMin Algorithm.	89
5.2	Expression of the Poissonized density.	90
5.2.1	Mean number of comparisons between two keys.	90
5.2.2	Mean number of comparisons between two parametrized words.	91
5.2.3	Density of an algorithm – A first expression.	91
5.2.4	Expression of the density in the Poisson model via its coefficients. Automatic transfer.	92
5.3	Exact mean number of comparisons.	94
5.3.1	Density and symbol-density.	94
5.3.2	Coefficients of the mean number of symbol comparisons in the Poisson model	94
5.3.3	Exact expression of the mean number of symbol-comparisons in the Bernoulli model.	95
5.4	Analytic study of the mixed Dirichlet series.	96
5.4.1	Using Rice method.	96
5.4.2	Tameness of the mixed Dirichlet series.	96
5.4.3	Tameness transfer.	97
5.4.4	Tameness of the mixed Dirichlet series.	97
5.4.5	Singular expression of the mixed Dirichlet series at $s = \sigma_0$.	98
5.4.6	Asymptotic expansions for $S(n)$.	98
5.5	Asymptotic study in the SelMin case.	98
5.5.1	Main result.	98
5.5.2	Study on the halfplane $\Re s \geq 2$.	99
5.5.3	Study on the vertical strip $\Re s \in [1, 2]$.	100

In this chapter, we describe our general framework for the realistic analysis of sorting and searching algorithms. We consider a (general) algorithm \mathcal{A} (which only performs comparisons and exchanges) and we study the mean number of *symbol comparisons* performed by the algorithm when the keys are *words* that are drawn from a given source \mathcal{S} (described in Chapter 2). This analysis is *more realistic* than the classical analyzes which study the mean number of keys comparisons in two ways. First it studies a more realistic cost, that is the number of symbol comparisons instead of the number of key comparisons. Second it deals with more realistic inputs: words drawn from a specified source instead of keys.

5.1 Main principles.

5.1.1 Two probabilistic models for the inputs of the algorithm.

The set of inputs is the probabilistic set of all sequences formed with infinite words independently produced by the same source \mathcal{S} . We denote by \mathcal{M} the set $\Sigma^{\mathbb{N}}$ when it is endowed with the probability defined by the source. As we deal with the parametrization of a source M defined in Chapter 2.4, the set \mathcal{M} is just $M(\mathcal{I})$ where the unit interval is endowed with the uniform probability. Then the probabilistic set of inputs of sorting and searching algorithms is then \mathcal{M}^* . This means the following: we consider a finite sequence $\mathcal{V} = (V_1, \dots, V_N) \in \mathcal{M}^*$ of infinite words independently produced by the same source \mathcal{S} . Such a sequence \mathcal{V} is obtained by N independent drawings v_1, v_2, \dots, v_N in the interval $\mathcal{I} = [0, 1]$ via the mapping M , and we set $V_i := M(v_i)$.

We recall (see Chapter 4) that there are two main probabilistic models of interest for the set \mathcal{M}^* of the inputs: the Poisson model and the Bernoulli model.

- (i) When the cardinality N of \mathcal{V} is fixed and equal to n , and words $V_i \in \mathcal{V}$ are independently emitted by the source \mathcal{S} , this is the (usual) Bernoulli model denoted by $(\mathcal{B}_n, \mathcal{S})$, and the equality $(\mathcal{B}_n, \mathcal{S}) = \mathcal{M}^n$ holds between the two probabilistic models.
- (ii) When the cardinality N of \mathcal{V} obeys a Poisson law of rate Z , that is

$$\mathbb{P}[N = k] = e^{-Z} \frac{Z^k}{k!}, \quad (5.1)$$

this is the Poisson model of rate Z , denoted by $(\mathcal{P}_Z, \mathcal{S})$, where there are two main steps:

- (a) the number N of words is drawn according to the Poisson law the rate Z ;
- (b) then, the N words are independently drawn from the source \mathcal{S} .

In the Poisson model, the rate Z plays a role much similar¹ to the cardinal of \mathcal{V} .

5.1.2 The cost of interest.

We want to study the random variable S defined as the number of symbol comparisons performed by the algorithm on a sequence of words independently drawn from the source. The random variable S is then defined on the set \mathcal{M}^* , and, by definition,

$S[\mathcal{V}] :=$ the number of symbol comparisons performed by the algorithm on the input \mathcal{V} .

¹In particular $\mathbb{E}_Z[N] = Z$ for a random variable N obeying a Poisson law of rate Z .

The average of the cost S on all sequences equivalent to \mathcal{V} up to some permutations is denoted as $S(\mathcal{V})$.

Our general approach follows the general approach described in Sections 4.1, 4.3 and 4.4 of Chapter 4.

- (i) Our final interest is the mean number $S(n)$ of symbol comparisons in the Bernoulli model $(\mathcal{B}_n, \mathcal{S})$, that is the expectation of the random variable S on the set \mathcal{M}^n .
- (ii) First we study the mean number of another random variable \hat{S} which counts the number of *normalized* symbol comparisons (to be described soon) in the Bernoulli model $(\mathcal{B}_n, \mathcal{S})$. This expectation is noted $\hat{S}(n)$, and as stated in Section 5.3.3, $S(n)$ is easily derived from $\hat{S}(n)$.
- (iii) It is easier to begin the study in the Poisson model $(\mathcal{P}_Z, \mathcal{S})$, where \hat{S}_Z denotes the mean number of normalized symbol comparisons, that is the expectation of the variable \hat{S} in the Poisson model. We *prove* that the coefficients $\varphi(k)$ of the series expansion of \hat{S}_Z can be computed in an explicit way.
- (iv) Chapter 4 explains that it is now possible to return in the Bernoulli model $(\mathcal{B}_n, \mathcal{S})$, and, as recalled in Section 5.3.3, the mean number $\hat{S}(n)$ is expressed as an explicit binomial sum

$$\hat{S}(n) = \sum_{k=1+\sigma_0}^n (-1)^k \binom{n}{k} \varphi(k), \quad \text{for some integer } \sigma_0,$$

and we easily obtain an expression for $S(n)$.

- (v) We *prove* that the sequence $\varphi(k)$ admits an analytic lifting $\varpi(s)$ for $\Re s > \sigma_0$ and we wish to use Rice method. This is possible provided that the function $\varpi(s)$ be tame at σ_0 , as defined in Section 4.3.4. We *prove* that it is the case, provided that the source itself be tame, as described in Chapter 3. We *state* such tameness conditions on the source. We then continue as described in Sections 4.3 and 4.4.

5.1.3 A general scheme for the analysis.

Like many studies in analytic combinatorics, such an analysis is divided into two main parts: an *algebraic* or *combinatorial* part and an *asymptotic* and *analytic* step. In analytic combinatorics, the first part computes a mathematical object (most of the times a generating function) with algebraic and exact tools, and the second part deals with the generating function as a function of the complex variable, studies its singularities, and transfers this knowledge about its singularities into an asymptotic expansion.

We adopt and adapt the same philosophy here. First we compute the generating function related to our specific problem: this is the *mixed Dirichlet series* $\varpi(s)$, that encapsulates both the properties of the source and the characteristics of the algorithm. In the algebraic part, it yields the *exact value of the mean number of symbol comparisons* performed by each algorithm on words produced by a parametrized source. We then perform a second part of the analysis where we study the analytic properties of the mixed Dirichlet series $\varpi(s)$, namely the position and the nature of its singularities, and transfer this knowledge to obtain the *asymptotic mean number of symbol comparisons*.

We deal with all the tools that were described in the previous chapters, and we detail more precisely how they are used.

5.1.4 Four steps in the analysis.

Both the algebraic analysis and the asymptotic analysis are themselves divided into two steps.

For technical reasons, that will be explained later in Section 5.2.1, we deal with a normalized cost \hat{S} which arises in a natural way in computations, even though it has not an actual algorithmic meaning.

First step of the algebraic analysis. We recall the set of keys \mathcal{M} is mapped to the unit interval. This step is devoted to computing the *density* of the algorithm. It uniquely depends on the algorithm and provides a *precise* measure of the mean number of *key-comparisons* performed near *specific points*. As it is easier to compute this density in the *Poisson model*, where the number of keys instead of being fixed follows a Poisson law of parameter Z , this first step provides an expression of the Poissonized density relative to each algorithm. We use the following chain to get the result. We begin with the mean number of local key-comparisons between two keys, obtained in Chapter 1. Then keys are viewed as words, and finally we use the parametrization of sources described in Section 2.4, and basic properties of the Poisson model, described in Section 4.1.

Second step of the algebraic analysis. Then, the source intervenes via its coincidence defined in Section 2.1.3, and its geometry defined by *fundamental triangles* (see Paragraph 2.4.3), which describe the location of pairs of words which share a common prefix. With integrals, we then obtain an exact expression for the mean number of symbol comparisons in the Poisson model. As described in Chapter 4, we deduce an exact expression of the mean number of symbol comparisons in the Bernoulli model where the number n of keys is fixed. This expression is given as a binomial sum which deals with the *mixed Dirichlet series*.

First step of the asymptotic analysis. We approach the corresponding asymptotic analysis with Rice method, described in Sections 4.3 and 4.4. To use this method, we need the mixed Dirichlet series to be tame (see Section 4.3.4). This series $\varpi(s)$ is obtained by *twisting* the generating function of the source $\Lambda(s)$. This twisting is induced by the algorithm and first depends on the type of the algorithm: is it a sorting algorithm or a searching algorithm? Then we need the source itself to be tame, in various senses related to Definitions given in Section 3.2. We prove that convenient tameness of the source indeed entails tameness of the mixed Dirichlet series.

Second Step of the asymptotic analysis. Finally, Proposition 4.8 provides asymptotic estimates for the mean number of symbol comparisons of the algorithm when it deals with words produced by a given source. Constants in these estimates will depend on the pair algorithm/source considered.

5.1.5 Main objects in the analysis.

Figure 5.1 describes the main objects which intervene in the two steps of the algebraic analysis. All the intermediary computation steps deal with the cost \hat{S} . In the table of Figure 5.1, all the lines (except the first line and the last line) deal with this normalized cost. In the table of Figure 5.2, all the lines (except the last line) deal with this normalized cost.

Objects of the analysis	Definition	Computation
$\pi_n(i, j)$	Mean number of key comparisons between the keys of rank i and j	Fig. 5.1 (Chap. 1)
$\hat{\pi}_n(i, j)$	Mean number of normalized key comparisons between the keys of rank i and j	Fig. 5.1 (Chap. 1), Eq. (5.2)
$\hat{\pi}[u, t]$	Mean number of normalized key comparisons between the words of parameters u and t	Eq. (5.4)
$\Phi_Z(u, t)$	Density of the algorithm at (u, t) in the model \mathcal{P}_Z	Eq. (5.6)
$\varphi(k, u, t)$	k -th coefficient of $\Phi_Z(u, t)$, i.e., $\varphi(k, u, t) = (-1)^k k! [Z^k] \Phi_Z(u, t)$	Prop. 5.5
\hat{S}_Z	Mean number of normalized symbol comparisons in the model \mathcal{P}_Z	Eq. (5.12)
$\varphi(k)$	k -th Coefficient of \hat{S}_Z , i.e., $\varphi(k) = (-1)^k k! [Z^k] S_Z$	Eq. (5.16)
$\hat{S}(n)$	Exact mean number of normalized symbol comparisons in the model \mathcal{B}_n	Eq. (5.17)
$S(n)$	Exact mean number of symbol comparisons in the model \mathcal{B}_n	Eq. (5.17)

Figure 5.1: The two steps of the *algebraic analysis*, the main objects used in the analysis, with their name, their definition and their computation.

Objects of the analysis	Definition and Properties	Method of analysis
$\varpi(s)$	Expression of the mixed Dirichlet series obtained as an analytic lifting of the sequence $k \mapsto \varphi(k)$ in $\Re s > \sigma_0$ <ul style="list-style-type: none"> – Relation of $\varpi(s)$ with Dirichlet series of the source – Singular expansion of $\varpi(s)/(s - \sigma_0)$ at $s = \sigma_0$ – Tameness of $\varpi(s)$ at $s = \sigma_0$ 	
$\hat{S}(n)$	Asymptotic expansion of the mean number of normalized symbol comparisons in the model \mathcal{B}_n <ul style="list-style-type: none"> – Dominant and sub-dominant terms, <ul style="list-style-type: none"> with the singular expansion of $\varpi(s)/(s - \sigma_0)$ – Remainder terms, with the tameness of $\varpi(s)$ 	Rice formula Rice formula
$S(n)$	Asymptotic expansion of the mean number of symbol comparisons in the model \mathcal{B}_n	Rice formula

Figure 5.2: The two steps of the *asymptotic analysis*, the main objects used in the analysis with their name, their definition and their computation.

5.1.6 General Results. The case of the SelMin Algorithm.

We describe a general framework which will be used and instantiated in the following two chapters. We explain the general approach, announce and state the results, but we do not prove them here. The following two chapters are devoted to apply and prove these general results to each studied algorithm. The notion of “studied algorithm” refers to all algorithms which will fall in our scrutiny in one of the following two chapters.

However, we choose to provide a first example in this chapter, and we precisely describe the analysis of the **SelMin** Algorithm (selection of the minimum of an array). This algorithm is different from algorithms studied in the next chapters, as this is a searching algorithm which does not belong to the **Quickselect** class. This analysis is interesting *per se* but it is also used as a connecting thread for giving this Chapter a more concrete feel to the reader.

5.2 Expression of the Poissonized density.

5.2.1 Mean number of comparisons between two keys.

We begin with the expressions of the mean numbers of local comparisons that we have studied in Chapter 1.

MetaTheorem 5.1. *All the expressions of $\pi_n(i, j)$ are rational fractions in $i, j, j - i$. For the algorithms *InsSort* and *BubSort*, $\pi_n(i, j)$ contains a constant term equal to $1/2$.*

Note. We have already given some intuition p. 90 in Eq. (5.1) why rational functions may indeed be expected.

Amongst these algorithms (and more generally amongst the sorting and searching algorithms), there are algorithms which perform a *systematic* comparison between keys that are in the *reverse* order. We say that these algorithms are *non-normalized*. Here, amongst the studied algorithms of Chapter 1, there are only two *non-normalized* algorithms, namely, the *InsSort* and the *BubSort* Algorithms. Non-normalized algorithms may perform several comparisons between two keys in reverse order, but we only consider the first comparison of this type², and we define it as *the* non-normalized comparison. In summary, a comparison is *non-normalized* if this is a systematic comparison performed for the first time. Otherwise, it is said to be normalized. And we define \widehat{K} , \widehat{S} as the total number of *normalized* key comparisons or *normalized symbol* comparisons.

For non-normalized algorithms, and in the permutation model, the non-normalized comparison arises with probability $1/2$, the probability that any two keys are in the reverse order. When the term $\pi_n^-(i, j)$ (associated with comparisons between keys in reverse order) contains a constant term $1/2$, this means that this algorithm is non-normalized, and we denote by $\widehat{\pi}_n(i, j)$ the term $\pi_n(i, j)$ from which this constant is removed, namely

$$\widehat{\pi}_n(i, j) := \begin{cases} \pi_n(i, j) - 1/2 & \text{for non-normalized algorithms} \\ \pi_n(i, j) & \text{for normalized algorithms.} \end{cases} \quad (5.2)$$

Then the mean number $\widehat{K}(n)$ is the sum of $\widehat{\pi}_n(i, j)$ for $1 \leq i < j \leq n$, and thus the following equality holds

$$K(n) = \begin{cases} \widehat{K}(n) + \frac{1}{2} \binom{n}{2} & \text{for non-normalized algorithms ,} \\ \widehat{K}(n) & \text{for normalized algorithms .} \end{cases}$$

In the same vein, the mean number $\widehat{S}(n)$ of normalized symbol comparisons is related to the mean number $S(n)$ of symbol comparisons by the equality

$$S(n) = \begin{cases} \widehat{S}(n) + \frac{1}{2} \binom{n}{2} c(\mathcal{S}) & \text{for non-normalized algorithms ,} \\ \widehat{S}(n) & \text{for normalized algorithms .} \end{cases} \quad (5.3)$$

where $c(\mathcal{S})$ is the coincidence of the source. $(c(\mathcal{S}) + 1)$ is the mean number of symbol comparisons needed to distinguish two random words.

For SelMin. *In this case, one has*

$$\widehat{\pi}_n(i, j) = \pi_n(i, j) = \frac{1}{i(i+1)} + \frac{1}{j(j-1)}.$$

This is a normalized algorithm.

²This is important as *BubSort* may perform subsequent comparisons between these two keys

5.2.2 Mean number of comparisons between two parametrized words.

Consider a sequence $\mathcal{V} \in \mathcal{M}^*$ that contains two given words $M(u)$ and $M(t)$. We define $\hat{\pi}[u, t][\mathcal{V}]$ as the number of (normalized) key comparisons performed by the algorithm on the input sequence \mathcal{V} between the keys $M(u)$ and $M(t)$, and by $\hat{\pi}[u, t]\langle\mathcal{V}\rangle$ the *mean* of the cost taken with respect to all the permutations of \mathcal{V} . We recall that $N_{\mathcal{J}}$ is a variable defined on \mathcal{M}^* as the number of words whose parameter belongs to \mathcal{J} . The following equalities hold

$$\text{Rank}_{\mathcal{V}} M(u) = N_{[0, u[}[\mathcal{V}] + 1, \quad \text{Rank}_{\mathcal{V}} M(t) = N_{[0, u[}[\mathcal{V}] + N_{]u, t[}[\mathcal{V}] + 2,$$

where the respective translations of 1 and 2 express that $M(u)$ and $M(t)$ belong to \mathcal{V} .

We now relate the mean number $\hat{\pi}[u, t]\langle\mathcal{V}\rangle$ to the (normalized) number $\hat{\pi}_N(i, j)$ defined in the previous Section 5.2.1. More precisely, the equality holds

$$\hat{\pi}[u, t]\langle\mathcal{V}\rangle = \hat{\pi}_{|\mathcal{V}|}(N_{[0, u[}[\mathcal{V}] + 1, N_{[0, u[}[\mathcal{V}] + N_{]u, t[}[\mathcal{V}] + 2). \quad (5.4)$$

We remark that the right side of (5.4) is defined on the total set \mathcal{M}^* . Setting $N = N_{[0, 1]}$, the function

$$\hat{\pi}_N(N_{[0, u[} + 1, N_{[0, u[} + N_{]u, t[} + 2). \quad (5.5)$$

provides thus an extension of the function $\mathcal{V} \mapsto \hat{\pi}[u, t]\langle\mathcal{V}\rangle$ on the total set \mathcal{M}^* (defined even for sequences which do not contain $M(u)$ or $M(t)$). This expression plays a central role in our analyses. In our framework, expressions obtained for $\hat{\pi}_N(i, j)$ ensure that $\hat{\pi}[u, t]$ is always a sum of rational functions in variables $N_{[0, u[}$ and $N_{]u, t[}$. Note that, in the Poisson model $(\mathcal{P}_Z, \mathcal{S})$, the variables $N_{[0, u[}$ and $N_{]u, t[}$ are themselves independent Poisson variables of parameters Zu and $Z(t - u)$ (respectively). This implies that the sum $N_{[0, u[} + N_{]u, t[}$ is also a Poisson variable of parameter Zt .

5.2.3 Density of an algorithm – A first expression.

We first give a formal definition of the density.

Definition 5.2. *For a given algorithm, and a parametrized source \mathcal{S} , the Poissonized density $\Phi_Z(u, t)$ at the point (u, t) of the triangle \mathcal{T} is the mean number in the Poisson model $(\mathcal{P}_Z, \mathcal{S})$ of (normalized) key comparisons performed by the algorithm between a pair $(M(u'), M(t'))$ with (u', t') close to (u, t) . More formally*

$$\Phi_Z(u, t) du dt := \text{the mean number of (normalized) key comparisons performed by the algorithm between } M(u') \text{ and } M(t') \text{ for } u' \in [u - du, u] \text{ and } t' \in [t, t + dt].$$

It is possible to relate the Poissonized density and the expectation of the random variable defined in (5.5). This is due to the nice properties of this model, and this is why we choose the Poisson model as our (first) probabilistic model.

Lemma 5.3. *The Poissonized density $\Phi_Z(u, t)$ satisfies*

$$\Phi_Z(u, t) = Z^2 \cdot \mathbb{E}_Z [\hat{\pi}_N(N_{[0, u[} + 1, N_{[0, u[} + N_{]u, t[} + 2)]]. \quad (5.6)$$

Proof. We consider the interesting subsets $K \in \text{Set}[\mathcal{M}]$ that contain two words of which the parameters belong respectively to $[u - du, u]$ and $[t, t + dt]$:

$$K \cap M[u - du, u] \neq \emptyset, \quad K \cap M[t, t + dt] \neq \emptyset,$$

and the subset $\mathcal{K} \subset \text{Set}[\mathcal{M}]$ which gathers such interesting subsets. We also consider the set $\underline{\mathcal{K}} \subset \mathcal{M}^*$ of all the sequences \underline{K} built from \mathcal{K} with all the possible permutations. The

subset $\underline{\mathcal{K}}$ gathers sequences which contain a pair $(M(u'), M(t'))$ with $u' \in [u - du, u]$ and $t' \in [t, t + dt]$. We wish to consider the restriction of the (normalized) cost \hat{S} to $\underline{\mathcal{K}}$, and the mean value which defines the density is exactly the expectation

$$\Phi_Z(u, t) du dt := \mathbb{E}_Z[\mathbf{1}_{\underline{\mathcal{K}}} \cdot \hat{S}] = \int_{\mathcal{M}^*} \mathbf{1}_{\underline{\mathcal{K}}}[\mathcal{V}] S[\mathcal{V}] d_Z \mathcal{V}.$$

Gathering all the sequences \mathcal{V} corresponding to the same set \mathcal{U} gives rise to the expression

$$\Phi_Z(u, t) du dt = \int_{\text{Set}[\mathcal{M}]} \mathbf{1}_{\mathcal{K}} \langle \mathcal{U} \rangle S \langle \mathcal{U} \rangle d_Z \mathcal{U}.$$

Now, when $\mathcal{U} \in \mathcal{K}$, it contains an interesting pair, and the equality holds, from the previous Section,

$$S \langle \mathcal{U} \rangle = \hat{\pi}_{|\mathcal{U}|}(N_{[0, u[}[\mathcal{U}] + 1, N_{[0, u[}[\mathcal{U}] + N_{]u, t[}[\mathcal{U}] + 2).$$

Remark that this last variable only depends on the intervals $[0, u[$ and $]u, t[$, whereas $\mathbf{1}_{\mathcal{K}}$ only depends on the intervals $[u - du, u]$ and $[t, t + dt]$. Then, in the Poisson model, the two functions are independent. We may moreover return from $\text{Set}(\mathcal{M})$ to \mathcal{M}^* , and finally get

$$\Phi_Z(u, t) du dt = \left(\int_{\mathcal{M}^*} \mathbf{1}_{\underline{\mathcal{K}}}[\mathcal{V}] d_Z \mathcal{V} \right) \cdot \left(\int_{\mathcal{M}^*} \hat{\pi}_{|\mathcal{V}|}(N_{[0, u[}[\mathcal{V}] + 1, N_{[0, u[}[\mathcal{V}] + N_{]u, t[}[\mathcal{V}] + 2) d_Z \mathcal{V} \right).$$

As the two intervals $[u - du, u]$ and $[t, t + dt]$ are disjoint, with the independence property of the Poisson model, the first integral equals

$$\mathbb{E}_Z[\mathbf{1}_{\underline{\mathcal{K}}}] = \mathbb{P}_Z[\underline{\mathcal{K}}] = Z du \cdot Z dt = Z^2 du dt.$$

The second integral is just $\mathbb{E}_Z[\hat{\pi}_N(N_{[0, u[} + 1, N_{[0, u[} + N_{]u, t[} + 2)]$. The lemma is proven \square

In order to use the Rice method of Section 4.3.5 we are interested in the coefficients of the following series expansion of $\Phi_Z(u, t)$ defined as

$$\Phi_Z(u, t) = \sum_{k \geq 2} (-1)^k \frac{Z^k}{k!} \varphi(k, u, t), \quad \varphi(k, u, t) := (-1)^k k! [Z^k] \Phi_Z(u, t) \quad (5.7)$$

and, with (5.6), the coefficients $\varphi(k, u, t)$ satisfy $\varphi(k, u, t) = 0$ for $k = 0, 1$. We now explain why these coefficients $\varphi(k, u, t)$ may be computed in an “automatic way”.

5.2.4 Expression of the density in the Poisson model via its coefficients. Automatic transfer.

We state our first result:

MetaTheorem 5.4. *For any studied algorithm, the random variable $\hat{\pi}[u, t]$ is expressed as a linear combination of basic random variables $G_m(N_{\mathcal{J}})$ with*

$$G_m(X) := \frac{1}{(X + 1)(X + 2) \dots (X + m)} \quad (m \geq 1), \quad (5.8)$$

and \mathcal{J} is an interval $\mathcal{J} \subset [0, 1]$.

For SelMin. In this case, the random variable $\hat{\pi}[u, t]$ is

$$\hat{\pi}[u, t] = \frac{1}{(N_{[0, u[} + 1)(N_{[0, u[} + 2)} + \frac{1}{(N_{[0, t[} + 1)(N_{[0, t[} + 2)}.$$

It decomposes in the basis G_m as $\hat{\pi}[u, t] = G_2(N_{[0, u[}) + G_2(N_{[0, t[})$.

The following proposition describes how to compute the expectations of the basic random variables in the Poisson model.

Proposition 5.5. Consider an integer $m \geq 1$ and an interval $\mathcal{J} \subset \mathcal{I} = [0, 1]$. The expectation in the Poisson model $(\mathcal{P}_Z, \mathcal{S})$ of the random variable $G_m(N_{\mathcal{J}})$ only depends on m and λZ where λ is the length of the interval \mathcal{J} . It is denoted by $F_m(\lambda Z)$. Moreover, the two sequences

$$\beta_m(k, \lambda) = (-1)^k k! [Z^k] (Z^2 F_m(\lambda Z)), \quad \gamma_m(k, \lambda) := (-1)^k k! [Z^k] (Z^3 F_m(\lambda Z))$$

admit the following expressions, resp. for $k > 1$ and $k > 2$,

$$\beta_m(k, \lambda) = \frac{1}{(m-1)!} \frac{k(k-1)}{k+m-2} \lambda^{k-2}, \quad \gamma_m(k, \lambda) = \frac{-1}{(m-1)!} \frac{k(k-1)(k-2)}{k+m-3} \lambda^{k-3}. \quad (5.9)$$

Proof. With Lemma 4.1, we know that $N_{\mathcal{J}}$ follows a Poisson law of parameter λZ . Then, $\mathbb{E}_Z[G_m(N_{\mathcal{J}})]$ only depends on m and λZ and

$$\mathbb{E}_Z[G_m(N_{\mathcal{J}})] = \sum_{k \geq 0} G_m(k) \mathbb{P}_Z[N_{\mathcal{J}} = k] = e^{-\lambda Z} \sum_{k \geq 0} G_m(k) \frac{(\lambda Z)^k}{k!} := F_m(\lambda Z).$$

We first compute the coefficients $\alpha_m(k, \lambda)$ in the series expansion of $F_m(Z)$

$$\alpha_m(k) := (-1)^k k! [Z^k] F_m(Z) = \frac{1}{(m-1)!} \frac{1}{k+m}. \quad (5.10)$$

Then, the coefficients $\beta_m(k, \lambda)$ and $\gamma_m(k, \lambda)$ are related to $\alpha_m(k)$, respectively for $k > 1$ and $k > 2$,

$$\beta_m(k, \lambda) = k(k-1) \lambda^{k-2} \alpha_m(k-2), \quad \gamma_m(k, \lambda) = -k(k-1)(k-2) \lambda^{k-3} \alpha_m(k-3),$$

which proves, with the help of (5.10), the expressions in Eq. (5.9) and finally the result. \square

With Theorem 5.4, and Proposition 5.5, we obtain our second Theorem.

MetaTheorem 5.6. For any studied algorithm, there exists an integer σ_0 , for which, for any $k \geq \sigma_0 + 1$, the sequence of coefficients $(\varphi(k, u, t))$ of the density $\Phi_Z(u, t)$ is expressed in a unique way as a linear combination of the sequences $(\beta_m(k, \lambda))$ and $(\gamma_m(k, \lambda))$ (defined in Eq. (5.9)) for various values of $\lambda \in]0, 1[$ and $m \in \mathbb{N}$. The precise linear combination and the choice of the various parameters depends on the algorithm. The integer σ_0 is equal to 1 or 2, and is equal to 2 only if the linear combination involves a sequence of type γ .

For SelMin. In this case, the integer σ_0 equals 1, and the coefficient $\varphi(k, u, t)$ satisfies

$$\varphi(k, u, t) = \beta_2(k, t) + \beta_2(k, u) = (k-1)[u^{k-2} + t^{k-2}]. \quad (5.11)$$

5.3 Exact mean number of comparisons.

In the model $(\mathcal{P}_Z, \mathcal{S})$, the density Φ_Z is a main tool for computing, not only the mean number of (normalized) key comparisons \widehat{K}_Z performed by the algorithm, but also the mean number of (normalized) symbol comparisons \widehat{S}_Z .

5.3.1 Density and symbol-density.

With the definition of density $\Phi_Z(u, t)$ given in Definition 5.2, the mean number of normalized key comparisons \widehat{K}_Z is obtained via the integral

$$\widehat{K}_Z = \mathcal{L}[\Phi_Z], \quad (5.12)$$

where \mathcal{L} is a linear functional defined for a function $\Phi : \mathcal{T} \rightarrow \mathbb{R}$ as

$$\mathcal{L}[\Phi] = \int_{\mathcal{T}} \Phi(u, t) du dt.$$

The definitions of the density $\Phi_Z(u, t)$ and the coincidence given in Definition 2.18 allow to define *symbol-density* as the product $[\gamma(u, t) + 1] \Phi_Z(u, t)$. This is the mean number of *normalized symbol comparisons* (in the Poisson model \mathcal{P}_Z) between two words $M(u')$ and $M(t')$ for (u', t') close to (u, t) . Then, the mean number of symbol comparisons \widehat{S}_Z is obtained via the formula

$$\widehat{S}_Z = \mathcal{J}[\Phi_Z] \quad \text{with} \quad \mathcal{J}[\Phi] = \int_{\mathcal{T}} [\gamma(u, t) + 1] \Phi(u, t) du dt. \quad (5.13)$$

As we have already seen in Chapter 2 (see Eq. (2.17)), the functional \mathcal{J} admits an alternative expression which involves the fundamental triangles \mathcal{T}_w ,

$$\mathcal{J}[\Phi_Z] := \int_{\mathcal{T}} (\gamma(u, t) + 1) \Phi_Z(u, t) du dt = \sum_{w \in \Sigma^*} \int_{\mathcal{T}_w} \Phi_Z(u, t) du dt. \quad (5.14)$$

This is a general phenomenon: formulas for the mean number of key comparisons or symbol comparisons are similar. When considering symbol comparisons, the functional \mathcal{J} replaces the simple integral \mathcal{L} used for key comparisons.

5.3.2 Coefficients of the mean number of symbol comparisons in the Poisson model

We now wish deal with the coefficients $\varphi(k)$ in the series expansion of \widehat{S}_Z , defined via the equality

$$\widehat{S}_Z = \sum_{k \geq 1 + \sigma_0} (-1)^k \frac{Z^k}{k!} \varphi(k). \quad (5.15)$$

Using the series expansion stated in (5.7), the expressions of \widehat{S}_Z as integrals (given in (5.12) and (5.14)) and the linearity of such integrals, the sequence $\varphi(k)$ is now defined, for any $k \geq 1 + \sigma_0$, in terms of integrals,

$$\varphi(k) := \int_{\mathcal{T}} (\gamma(u, t) + 1) \varphi(k, u, t) du dt = \sum_{w \in \Sigma^*} \int_{\mathcal{T}_w} \varphi(k, u, t) du dt. \quad (5.16)$$

Theorem 5.6 provides a close expression of the coefficients $\varphi(k, u, t)$. It is then easy to obtain the expression of the coefficients $\varphi(k)$ of the mean number S_Z via computations of the integrals of $\varphi(k, u, t)$ on the triangles \mathcal{T}_w . This leads to

MetaTheorem 5.7. *For any studied algorithm, there exists an integer $\sigma_0 \in \{1, 2\}$, for which the coefficients $\varphi(k)$ of the mean number of normalized symbol comparisons \hat{S}_Z in the Poisson model admit an explicit expression for $k > \sigma_0$. The sequence $\varphi(k)$ admits an analytic lifting on for $\Re s > \sigma_0$.*

For SelMin. *In this case, the coefficient $\varphi(k)$ is equal to*

$$\varphi(k) = \sum_{w \in \Sigma^*} (b_w - a_w) \left[b_w^{k-1} - a_w^{k-1} \right].$$

Proof. We take the integral of $\varphi(k, u, t)$ of (5.11) on each fundamental triangle \mathcal{T}_w , we obtain:

$$\begin{aligned} \frac{\varphi(k)}{k-1} &= \sum_{w \in \Sigma^*} \int_{(u,t) \in \mathcal{T}_w} (u^{k-2} + t^{k-2}) du dt \\ &= \sum_{w \in \Sigma^*} \int_{a_w}^{b_w} u^{k-2} (u - a_w) du, + \sum_{w \in \Sigma^*} \int_{a_w}^{b_w} u^{k-2} (b_w - u) du \\ &= \sum_{w \in \Sigma^*} (b_w - a_w) \int_{a_w}^{b_w} u^{k-2} du, \end{aligned}$$

so that one has

$$\varphi(k) = \sum_{w \in \Sigma^*} (b_w - a_w) \left(b_w^{k-1} - a_w^{k-1} \right).$$

□

5.3.3 Exact expression of the mean number of symbol-comparisons in the Bernoulli model.

We now wish to return to the Bernoulli model $(\mathcal{B}_n, \mathcal{S})$, where the number of keys is fixed and equal to n . The mean number $\hat{S}(n)$ of normalized symbol comparisons used by the algorithm when it deals with n words independently drawn from the same source is related to \hat{S}_Z and then to the sequence $\varphi(k)$ by the equation

$$\hat{S}(n) = \sum_{k=1+\sigma_0}^n (-1)^k \binom{n}{k} \varphi(k). \quad (5.17)$$

As the coefficient $\varphi(k)$ are explicit, as stated in Theorem 5.7, this provides an explicit formula for $\hat{S}(n)$. Now, we return to the “true” cost S , and with (5.3)

$$S(n) = \begin{cases} \hat{S}(n) + \frac{\Lambda(2)}{2} \binom{n}{2} & \text{for InsSort and BubSort} \\ \hat{S}(n) & \text{for the other algorithms.} \end{cases}$$

MetaTheorem 5.8. *For any studied algorithm, there exists an integer $\sigma_0 \in \{1, 2\}$, and a sequence $\varphi(k)$ defined for $k \geq 1 + \sigma_0$ for which the mean number of symbol comparisons $S(n)$ in the Bernoulli model is expressed as a binomial sum of the form*

$$S(n) = \begin{cases} \sum_{k=1+\sigma_0}^n (-1)^k \binom{n}{k} \varphi(k) + \frac{\Lambda(2)}{2} \binom{n}{2} & \text{for InsSort and BubSort} \\ \sum_{k=1+\sigma_0}^n (-1)^k \binom{n}{k} \varphi(k) & \text{for the other algorithms.} \end{cases}$$

Moreover, the sequence $\varphi(k)$ is explicit and admits an analytic lifting³ $\varpi(s)$ on the halfplane $\Re s > \sigma_0$.

For SelMin. In this case, the mean number of symbol comparisons is

$$S(n) = \sum_{w \in \Sigma^*} (b_w - a_w) \sum_{k=2}^n (-1)^k \binom{n}{k} \left[b_w^{k-1} - a_w^{k-1} \right].$$

We have here concluded the algebraic part of the analysis and we have now the exact expression of $S(n)$. It is obtained in an “automatic” way, from the expectations $\pi_n(i, j)$ given in Chapter 1, using the geometry of the source described in Chapter 2, and the general Poisson-Bernoulli framework described in Section 4.1

5.4 Analytic study of the mixed Dirichlet series.

The algebraic analysis also provides an explicit expression of the analytic lifting $\varpi(s)$ on the halfplane $\Re s > \sigma_0$, where σ_0 is defined in Theorem 5.6.

For SelMin. In this case, the mixed series is defined for $\Re s > 1$ as

$$\varpi(s) = \sum_{w \in \Sigma^*} (b_w - a_w) \left[b_w^{s-1} - a_w^{s-1} \right].$$

5.4.1 Using Rice method.

We wish to use the Rice method presented in Section 4.3, that it is now recalled:

Assume that the lifting $\varpi(s)$ of the sequence $\varphi(k)$ is meromorphic in a region \mathcal{R} which contains the half plane $\Re s \geq \sigma_0$ and is of polynomial growth there (for $|\Im s| \rightarrow \infty$). Then

$$S(n) = - \left[\sum_k \text{Res}[\varpi(s) L_n(s); s_k] + \frac{1}{2i\pi} \int_{\mathcal{C}} \varpi(s) L_n(s) ds \right], \quad (5.18)$$

where \mathcal{C} is a curve positively oriented of class \mathcal{C}^1 included in \mathcal{R} and the sum is extended to all poles s_k of $\varpi(s) \cdot L_n(s)$ inside the domain \mathcal{D} which lies between the vertical line $\Re s = \sigma_0$ and the curve \mathcal{C} .

This application of Rice method is possible, *only if* the series $\varpi(s)$ is tame at σ_0 , as it is defined in Section 4.3.4 and if we have a good knowledge of its poles and their residues.

5.4.2 Tameness of the mixed Dirichlet series.

We first need to prove the tameness of the Dirichlet series $\varpi(s)$ at σ_0 . For this, we analyse the general form of $\varpi(s)$ and we compare it to one of the Dirichlet series of the source Λ or Π .

³The lifting coincides on values $s = k$ (see Section 4.3).

Relation between the mixed Dirichlet series and the Dirichlet series of the source.

MetaTheorem 5.9. *The form of the mixed Dirichlet depends on the type of the algorithm: sorting algorithm or searching algorithm.*

- (a) *For any of the three sorting algorithms, the mixed series $\varpi(s)$ is closely related to some variant $\Lambda[F](s)$ of the Dirichlet series $\Lambda(s)$, that involves a function $F : [0, 1]^2 \rightarrow \mathbb{R}^+$ of class \mathcal{C}^1*

$$\Lambda[F](s) := \sum_{w \in \Sigma^*} F(a_w, b_w) p_w^s, \quad (5.19)$$

where a_w and b_w are the ends of the fundamental intervals (i.e., $\mathcal{I}_w := [a_w, b_w]$ and $p_w = b_w - a_w$). More precisely, the equality holds

$$\varpi(s) = A(s)\Lambda[F](s - 1 + \sigma_0),$$

where $A(s)$ is a rational function of s .

- (b) *For searching algorithms, there exist relations between the mixed Dirichlet $\varpi(s)$ and the Dirichlet series Λ and/or Π of the source, but these are only inequalities.*

$$|\varpi(s)| \leq |P(s)| \Lambda(\sigma') \quad \text{or} \quad |\varpi(s)| \leq |P(s)| \Pi(\sigma''),$$

for some reals $\sigma' > 1$ and $\sigma'' > 0$ related to $\sigma := \Re s$, and some polynomial P .

This Theorem will be made more precise for each class of algorithms (sorting or searching).

5.4.3 Tameness transfer.

As the series $\varpi(s)$ is closely related to the Dirichlet generating functions of the source, we can transfer to the mixed Dirichlet series $\varpi(s)$ the various notions of tameness that we have defined in Chapter 3 for the source \mathcal{S} . In particular the Λ -tameness, largely described in Chapter 3, but also the weak-tameness (weak Λ -tameness or weak Π -tameness described in Section 3.3).

- (a) For sorting algorithms, there is an exact relation between $\varpi(s)$ and $\Lambda[F](s + 1 - \sigma_0)$. Then, the tameness of $\varpi(s)$ at $s = \sigma_0$ is related to the tameness of $\Lambda[F]$ at $s = 1$. Then the Λ -tameness of the source may be transferred to the tameness of $\varpi(s)$ with the same shape.
- (b) For searching algorithms, there are inequalities between $\varpi(s)$ and $\Lambda(\sigma')$ with $\sigma' > 1$, or between $\varpi(s)$ and $\Pi(\sigma'')$ with $\sigma'' > 0$. Then, the weak tameness of the source (weak Π -tameness, or weak Λ -tameness) may be transferred to the tameness of $\varpi(s)$ at $s = 1$, and this is a tameness with a S -shape.

5.4.4 Tameness of the mixed Dirichlet series.

MetaTheorem 5.10. *For any studied algorithm, tameness properties of the source \mathcal{S} described in Chapter 3 are sufficient to entail the Dirichlet mixed series $\varpi(s)$ to be tame at σ_0 . The convenient tameness properties of the source depend on the type of the algorithm: sorting algorithm or searching algorithm.*

For SelMin. *The tameness of $\varpi(s)$ at $s = 1$ will be studied in Section 5.5.*

5.4.5 Singular expression of the mixed Dirichlet series at $s = \sigma_0$.

It remains to study the possible singularities of $\varpi(s)$ at σ_0 . Due to the occurrence of Rice kernel, and as σ_0 is always an integer, the function of interest is the ratio $\varpi(s)/(s - \sigma_0)$. Again, there are two main cases, depending on the type of algorithm (sorting or searching).

MetaTheorem 5.11. *For any studied algorithm, the behaviour of $\varpi(s)$ at $s = \sigma_0$ depends on the type of the algorithm*

- (a) *For sorting algorithms, the function $\varpi(s)$ has a pole of order at least one at σ_0 , with a leading term which involves the entropy. Then the ratio $\varpi(s)/(s - \sigma_0)$ has a pole of order at least two at σ_0*
- (b) *For searching algorithms, the function $\varpi(s)$ is analytic at $s = \sigma_0$ and the ratio $\varpi(s)/(s - \sigma_0)$ has a pole of order 1. Moreover, the value of $\varpi(s)$ at $s = \sigma_0$ involves constants which are related to various notions of coincidence.*

5.4.6 Asymptotic expansions for $S(n)$.

With Theorems 5.10 and 5.11, together with Proposition 4.8, we obtain an asymptotic expansion for the mean number of symbol comparisons in the Bernoulli model:

MetaTheorem 5.12. *For any studied algorithm, as soon as the source satisfies tameness properties adapted to the class of algorithms, there is an asymptotic expansion for the mean number of symbol comparisons in the Bernoulli model. The dominant terms involve specific constants obtained by residue calculus from the singular expression of $\varpi(s)$, and the remainder terms are related to the tameness of the mixed Dirichlet series.*

5.5 Asymptotic study in the SelMin case.

We now perform the analytic study in the case of the Algorithm Selmin. We start with the expression of the function $\varpi(s)$ in this case,

$$\varpi(s) = (s - 1) \sum_{w \in \Sigma^*} (b_w - a_w) \int_{a_w}^{b_w} u^{s-2} du. \quad (5.20)$$

We know that σ_0 is equal to 1, and we wish to study its tameness and its singular expression at $s = 1$.

5.5.1 Main result.

Proposition 5.13. *Assume the source \mathcal{S} to be weakly tame. Then, the mixed Dirichlet series $\varpi(s)$ relative to SelMin algorithm is tame at $\sigma_0 = 1$ with order $k_0 = 0$ and a S -shape.*

- (i) *the singular expression of $\varpi(s)/(s - 1)$ is*

$$\frac{\varpi(s)}{s - 1} = a(\mathcal{S}) \frac{1}{s - 1} + o(s - 1),$$

and involves the constant $a(\mathcal{S})$ defined as the mean length of the maximal prefix w of the form $w = 0^k$ of a random word from \mathcal{M} .

- (ii) *the width δ depends on an exponent a , attached to the source and defined in (5.27).*

The following subsections are devoted to the proof of this proposition. There are three cases: the halfplane $\Re s \geq 2$, the vertical strip $\Re s \in [1, 2]$, and finally the left of the vertical line $\Re s = 1$.

5.5.2 Study on the halfplane $\Re s \geq 2$.

With the expression of $\varpi(s)$ in (5.20), there is an easy upper bound, valid for $\sigma \geq 2$,

$$|\varpi(s)| \leq |s-1| \sum_{w \in \Sigma^*} (b_w - a_w) \int_{a_w}^{b_w} u^{\sigma-2} du \leq |s-1| \Lambda(2),$$

that proves that $\varpi(s)$ is tame of order 0 for any σ_0 for $\sigma_0 \geq 2$. The sequel of the proof is devoted to the case $\sigma \leq 2$.

We first consider the Dirichlet series of depth ℓ , namely

$$\varpi_\ell(s) = (s-1) \sum_{w \in \Sigma^\ell} (b_w - a_w) \int_{a_w}^{b_w} u^{s-2} du,$$

where Σ^ℓ is the set of the prefixes of length ℓ .

The first term of the sum which defines ϖ_ℓ is particular. It is relative to the prefix $\alpha_\ell = 0^\ell$, whose probability is denoted by $q_\ell = p_{\alpha_\ell}$. So we have

$$(s-1)q_\ell \int_0^{q_\ell} u^{s-2} du = q_\ell^s.$$

We then consider the remainder of the sum

$$R_\ell(s) = \sum_{\substack{w \in \Sigma^\ell, \\ w > \alpha_\ell}} (b_w - a_w)(b_w^{s-1} - a_w^{s-1}),$$

and prove that the series of general term $R_\ell(s)$ is normally convergent for $\Re s > 1 - \delta_2$ for some $\delta_2 > 0$. Then, the sum $R(s)$ of this series defines a tame function at $\sigma_0 = 1$ of abscissa δ_2 . And the equality $R(1) = 0$ holds since on the real axis $R(s)$ is positive for $s > 1$ and negative for $s < 1$. Finally,

$$\varpi(1) = a(\mathcal{S}) = \sum_{\ell \geq 0} q_\ell, \quad \text{with } q_\ell := \text{probability of the prefix } w = 0^\ell. \quad (5.21)$$

We then study $R_\ell(s)$ for $\sigma \leq 2$. We consider a real $A \in [0, 1]$ (to be fixed later as a function of s and ℓ) and split the sum into three sums, each of them relative to a subset of prefixes: the prefixes w for which $b_w < A$, the prefixes w for which $a_w > A$ and finally the unique prefix β for which $A \in [a_\beta, b_\beta]$. These sums are respectively denoted by $R_\ell^{(-)}(s)$, $R_\ell^{(+)}(s)$, $R_\ell^{(=)}(s)$.

When $a_w > A$, we use the mean-value theorem. There exists $c_w \in [a_w, b_w]$ for which

$$|b_w^{s-1} - a_w^{s-1}| \leq |s-1| c_w^{\sigma-2} (b_w - a_w) \leq |s-1| A^{\sigma-2} (b_w - a_w),$$

and this entails the inequality

$$|R_\ell^{(+)}(s)| \leq |s-1| A^{\sigma-2} \Lambda_\ell(2). \quad (5.22)$$

There are two cases for the other two sums, according to the position of $\sigma := \Re s$ with respect to 1.

5.5.3 Study on the vertical strip $\Re s \in [1, 2]$.

We first consider the situation on the vertical strip $\Re s \in [1, 2]$. In this case, when $b_w < A$, we remark the inequality

$$|b_w^{s-1} - a_w^{s-1}| \leq |b_w^{s-1}| + |a_w^{s-1}| = b_w^{\sigma-1} + a_w^{\sigma-1} \leq 2A^{\sigma-1},$$

and this entails the inequality

$$|R_\ell^{(-)}(s)| \leq 2A^{\sigma-1} \cdot A = 2A^\sigma. \quad (5.23)$$

For the sum $R_\ell^{(=)}(s)$, we use the upper bound

$$|b_w^{s-1} - a_w^{s-1}| \leq 2, \quad R_\ell^{(=)}(s) \leq 2 p_\beta \leq 2\pi_\ell \leq 2\Lambda_\ell(2)^{1/2}, \quad (5.24)$$

which deals with the length p_β of the interval $[a_\beta, b_\beta]$ and where $\pi_\ell := \max\{p_w, w \in \Sigma^\ell\}$. Finally, with (5.22), (5.23), (5.24), one obtains when $1 \leq \sigma \leq 2$,

$$R_\ell(s) \leq A^{\sigma-2}|s-1|\Lambda_\ell(2) + 2A^\sigma + 2\Lambda_\ell(2)^{1/2}.$$

We choose the value of A so that the first two terms are equal, namely

$$2A^\sigma = |s-1| A^{\sigma-2} \Lambda_\ell(2), \quad \text{i.e.,} \quad A = \left(\frac{|s-1|}{2} \right)^{1/2} \Lambda_\ell(2)^{1/2}.$$

Then, rewriting $\Lambda_\ell(2)^{\sigma/2} = \Lambda_\ell(2)^{\sigma/2} \Lambda_\ell(1)^{1-\sigma/2}$ and using the well-known log-convexity bound of $s \mapsto \Lambda_\ell(\sigma)$ and the equality $\Lambda_\ell(1) = 1$, one has the inequality for $\sigma \leq 2$

$$\Lambda_\ell(2)^{\sigma/2} \leq \Lambda_\ell \left(2\frac{\sigma}{2} + 1 - \frac{\sigma}{2} \right) = \Lambda_\ell \left(1 + \frac{\sigma}{2} \right), \quad (5.25)$$

one has

$$\left| R_\ell^{(-)}(s) + R_\ell^{(+)}(s) \right| \leq 4 \left(\frac{|s-1|}{2} \right)^{\sigma/2} \Lambda_\ell(2)^{\sigma/2} \leq 4 \left(\frac{|s-1|}{2} \right)^{\sigma/2} \Lambda_\ell \left(1 + \frac{\sigma}{2} \right).$$

Finally, if the source is weakly tame, $R(s)$ is analytic and of polynomial growth for $\Re(s) \geq 1$.

5.5.4 Situation on the left of the vertical line $\Re s = 1$

We now consider the case where $\sigma \leq 1$. The upper bounds for $R_\ell^{(-)}(s)$ and $R_\ell^{(=)}(s)$ are

$$\left| R_\ell^{(-)}(s) \right| \leq 2q_\ell^{\sigma-1} A, \quad \left| R_\ell^{(=)}(s) \right| \leq 2q_\ell^{\sigma-1} \pi_\ell, \quad (5.26)$$

and involve the probability $\pi_\ell := \max\{p_w, w \in \Sigma^\ell\}$.

The upper bound for $|R_\ell^{(+)}(s)|$ is the same as previously in (5.22). We choose A such that the bounds, for $|R_\ell^{(-)}(s)|$ in (5.26), and $|R_\ell^{(+)}(s)|$ in (5.22), are equal

$$2q_\ell^{\sigma-1} A = |s-1| \Lambda_\ell(2) A^{\sigma-2}, \text{ i.e., } A = \left(\frac{|s-1|}{2} \right)^{1/(3-\sigma)} (\Lambda_\ell(2) q_\ell^{1-\sigma})^{1/(3-\sigma)}.$$

Then, one has

$$\left| R_\ell^{(-)}(s) + R_\ell^{(+)}(s) \right| \leq 4 \left(\frac{|s-1|}{2} \right)^{1/(3-\sigma)} (\Lambda_\ell(2) q_\ell^{1-\sigma})^{1/(3-\sigma)} \cdot q_\ell^{\sigma-1}.$$

To conclude in this case, we need to compare the two sequences q_ℓ and $\Lambda_\ell(2)$ which already satisfy $q_\ell \leq \Lambda_\ell(2)^{1/2}$ (since $q_\ell^2 \leq \Lambda_\ell(2)$) and we assume that q_ℓ is not too small with respect to $\Lambda_\ell(2)$, namely

$$a := \limsup \left| \frac{\log q_\ell}{\log \Lambda_\ell(2)} \right| < +\infty. \quad (5.27)$$

Then, for ℓ large enough, one has $q_\ell \geq \Lambda_\ell(2)^a$, and

$$\left| R_\ell^{(-)}(s) + R_\ell^{(+)}(s) \right| \leq |C(s)| \Lambda_\ell(2)^b, \quad \text{with } b = \frac{1}{(3-\sigma)} (1 + a(1-\sigma)(\sigma-2)),$$

where the function $C(s)$ is of polynomial growth. For the remainder term $R_\ell^{(=)}(s)$, one has, with (5.27) and (5.26),

$$\pi_\ell \leq \Lambda_\ell(2)^{1/2} \quad \text{so that} \quad \left| R_\ell^{(=)}(s) \right| \ll \Lambda_\ell(2)^c \quad \text{with } c = \frac{1}{2} + a(\sigma-1).$$

The two exponents b and c depend continuously on σ and equal $1/2$ for $\sigma = 1$. Then, for any $b_0 > 1/2 - \epsilon$, there exists $\delta_3 > 0$ such that b, c satisfy $b, c > b_0$ for $\sigma > 1 - \delta_3$. Then, the convexity argument already used in (5.25) proves that

$$|R_\ell(s)| \leq |C(s)| \Lambda_\ell(1 + b_0),$$

which defines a convergent series. Then, if the source is weakly tame, the related series $R(s)$ is analytic and of polynomial growth in the vertical strip $\sigma \in]1 - \delta_3, 1]$.

Finally, we have proven that the series $\varpi(s)$ is tame, with a S -shape, related to the half plane $\Re s > 1 - \delta_3$ for some $\delta_3 > 0$. Moreover, at $s = 1$, the series satisfies (5.21) and we obtain the “singular” expansion of $\varpi(s)$ at $s = \sigma_0 = 1$.

Conclusion for the Chapter. Our general method is now described, with its Theorems. It remains to prove them, for sorting algorithms in Chapter 6 and for **QuickSelect** in Chapter 7.

Chapter 6

Towards a realistic analysis of three sorting algorithms. Lower bounds for the digital cost.

Contents

6.1	The two main results	104
6.1.1	Tameness properties of the source.	104
6.1.2	The main result.	105
6.1.3	Main constants of interest.	105
6.1.4	Beyond Λ -tameness?	106
6.1.5	Robustness.	107
6.1.6	An asymptotic lower bound for $S(n)$.	107
6.2	Mean number of symbol comparisons.	107
6.2.1	Expression for $\pi_n(i, j)$	107
6.2.2	Expression of $\hat{\pi}[u, t]$ in the basis G_m .	107
6.2.3	Expression of the coefficient $\varphi(k, u, t)$	108
6.2.4	Expressions of the mixed Dirichlet series $\varpi(s)$.	108
6.2.5	Tameness of the mixed Dirichlet series.	109
6.2.6	Mean number of symbol comparisons	111
6.3	An alternative approach.	111
6.3.1	Seidel's point of view.	112
6.3.2	Seidel's result	112
6.3.3	Faithfulness.	113
6.3.4	A relation between faithful sorting algorithms and trie parameters.	114
6.3.5	Analysis of additive trie parameters.	116
6.3.6	Proof of Proposition 6.11.	117
6.3.7	An alternative proof for QuickSort and InsSort .	119
6.4	An asymptotic lower bound for $S(n)$.	120
6.4.1	Proof of Theorem 6.5.	120
6.4.2	Proof of Lemma 6.13	121
6.4.3	Relation between various lower bounds.	122

In this chapter, we apply the general method described in Chapter 5 to three sorting algorithms **QuickSort**, **InsSort**, **BubSort**. This leads to one of the main results of the thesis, stated as Theorem 6.2. Then, we focus on lower bounds for the mean number of symbol comparisons performed by any sorting algorithm dealing with the standard (lexicographic) word comparison. We first recall the approach due to Seidel together with the notion of faithfulness he introduced [65]. This approach naturally leads to the structure of trie, and we make a detour to the analysis of trie parameters. Combining the approach of Seidel and ours leads to a lower bound for the mean number of symbol comparisons of any sorting algorithm using words emitted by a Λ -tame source. This result, stated as Theorem 6.5, is the second main result of the thesis.

6.1 The two main results

6.1.1 Tameness properties of the source.

Tameness and the main analytical properties were introduced in Chapter 3 with respect to the source and its Dirichlet series. We define a slightly different notion of tameness which enables a more direct derivation of analytical properties from Dirichlet series of the source to $\varpi(s)$. We recall the definition of the Dirichlet series of the source

$$\Lambda(s) = \sum_{w \in \Sigma^*} p_w^s, \quad \Lambda_k(s) = \sum_{w \in \Sigma^k} p_w^s.$$

There are also extensions of $\Lambda(s)$ and $\Lambda_k(s)$ which involve the fundamental probabilities p_w , together with the ends a_w, b_w of the fundamental intervals (see Section 2.4), via a function $F : [0, 1]^2 \rightarrow \mathbb{R}^+$ of class \mathcal{C}^1 ,

$$\Lambda[F](s) := \sum_{w \in \Sigma^*} F(a_w, b_w) p_w^s, \quad \Lambda_k[F](s) := \sum_{w \in \Sigma^k} F(a_w, b_w) p_w^s. \quad (6.1)$$

For $F \equiv 1$, we recover the classical function $\Lambda := \Lambda[1]$. On the half-plane $\sigma := \Re s > 1$, these series satisfy the relation $|\Lambda[F](s)| \leq \|F\| \Lambda(\sigma)$, where the norm $\|\cdot\|$ is the sup-norm on $[0, 1] \times [0, 1]$.

We now describe properties of the source that will entail tameness for the mixed series $\varpi(s)$.

Definition 6.1 (Tameness of Sources). *Denote by \mathcal{F} the set of functions $F : [0, 1]^2 \rightarrow \mathbb{R}^+$ of class \mathcal{C}^1 . A source is Λ -tame if $\Lambda(s)$ admits at $s = 1$ a simple pole, with a residue equal to $1/h(\mathcal{S})$, (where $h(\mathcal{S})$ is the entropy of the source) and if for any $F \in \mathcal{F}$, there exists $X \in \{S, H, P\}$ (shorthand for strip, hyperbolic and periodic) for which the series $\Lambda[F]$ is tame at $s = 1$ with a X -shape.*

Remark. If $\Lambda(s)$ admits at $s = 1$ a simple pole, with a residue equal to $1/h(\mathcal{S})$, then any series $\Lambda[F](s)$ for any $F \in \mathcal{F}, F > 0$, admits at $s = 1$ a simple pole, with a residue equal to (see the end of the proof of Proposition 6.6 in Section 6.2.5 for details)

$$\frac{1}{h(\mathcal{S})} \int_0^1 F(x, x) dx.$$

6.1.2 The main result.

We describe here the final results about the asymptotic expression of $S(n)$ for three algorithms. All the proofs are described in Section 6.2.

Theorem 6.2. *Consider a general source \mathcal{S} , assumed to be Λ -tame. Then, the mean number $S(n)$ of symbol comparisons performed by each of the three algorithms on a sequence of n words independently drawn from the same source \mathcal{S} admits the asymptotic behavior described in Table of Figure 6.1.*

- (i) [Dominant terms.] *The constants $h(\mathcal{S})$ and $c(\mathcal{S})$ in the dominant terms are described in the next Proposition 6.3.*
- (ii) [Sub-dominant terms.] *The constants κ_i in the sub-dominant terms are described in the next Proposition 6.3.*
- (iii) [Error terms.] *The following holds*
 - (a) *if the source has a S -shape with width δ , then $E(n) = O(n^{1-\delta})$;*
 - (b) *if the source has a H -shape with exponent β_0 , then $E(n) = n \cdot O(\exp[-(\log n)^\beta])$ with $\beta < 1/(1 + \beta_0)$;*
 - (c) *if the source has a P -shape with width δ , then $E(n) = n \cdot \Phi(n) + O(n^{1-\delta})$ where $n \cdot \Phi(n)$ is the expansion given by the family of imaginary poles (s_k) of $\Lambda(s)$ on $\Re s = 1$.*

Algorithms	$K(n)$	Dominant term of $S(n)$	Sub-dominant terms of $S(n)$	Remainder term of $S(n)$ (see 6.2)
QuickSort	$2n \log n$	$\frac{1}{h(\mathcal{S})} n \log^2 n$	$\kappa_0 n \log n + \kappa_2 n$	$E(n)$
InsSort	$\frac{n^2}{4}$	$\frac{c(\mathcal{S})}{4} n^2$	$\frac{1}{h(\mathcal{S})} n \log n + \left(\kappa_0 - \frac{c(\mathcal{S})}{4}\right) n$	$E(n)$
BubSort	$\frac{n^2}{2}$	$\frac{1}{4h(\mathcal{S})} n^2 \log n$	$\left(\kappa_1 + \frac{c(\mathcal{S})}{4}\right) n^2$	$nE(n)$

Figure 6.1: Results for Theorem 6.2. Constants are described in Proposition 6.3.

6.1.3 Main constants of interest.

We finally describe the main constants which intervene in the dominant terms of the singular expression of $\varpi(s)/(s - \sigma_0)$ at $s = \sigma_0$.

Proposition 6.3. (a) *The constants which intervene in the dominant terms displayed in the third column of Figure 6.1 are:*

- (i) *The entropy $h(\mathcal{S})$ of the source (defined in Definition 2.2).*
- (ii) *The coincidence $c(\mathcal{S})$, namely the mean number of symbols needed to compare two random words produced by the source (see Lemma 2.5).*

The constants κ_i which intervene in the sub-dominant terms¹ displayed in the fourth column of Figure 6.1 involve the Euler constant γ together with the sub-dominant constant of the source $d(\mathcal{S})$ equal to the constant term in the singular expansion of $\Lambda(s)$ at $s = 1$,

$$d(\mathcal{S}) = \lim_{s \rightarrow 1} \left[\Lambda(s) - \frac{1}{h(\mathcal{S})} \frac{1}{s-1} \right],$$

under the form

$$\kappa_0 = \frac{2}{h(\mathcal{S})}(\gamma - 2) + 2d(\mathcal{S}), \quad \kappa_1 = \frac{1}{8h(\mathcal{S})}(2\gamma - 3) + \frac{d(\mathcal{S})}{4}.$$

The entropy $h(\mathcal{S})$ and the coincidence $c(\mathcal{S})$ are defined in Chapter 2. The constants $c(\mathcal{S})$ and $h(\mathcal{S})$ are easy to compute for any memoryless source. For the unbiased source \mathcal{M}_r , one has:

$$c(\mathcal{M}_r) = \frac{r}{r-1}, \quad h(\mathcal{M}_r) = \log r.$$

For the source \mathcal{B}_p on the alphabet $\{0, 1\}$, with $p := p_0$, one has

$$c(\mathcal{B}_p) = \frac{1}{2p(1-p)}, \quad h(\mathcal{B}_p) = -p \log p - (1-p) \log(1-p).$$

6.1.4 Beyond Λ -tameness?

In this chapter, we insist on sources which are Λ -tame, as they are the most natural. However, our results can be extended to other sources, whose Dirichlet series Λ fulfills more general tameness properties.

Proposition 6.4. *Consider a source \mathcal{S} and its Dirichlet series $\Lambda(s)$. The following holds:*

- (i) *If the Dirichlet series Λ is tame at $s = s_0$, with $s_0 \in]1, 2[$ and order 1, then the asymptotic order of the mean number $S(n)$ of symbol comparisons performed by each of the three algorithms is described in the second column of Figure 6.2.*
- (ii) *If the Dirichlet series is Λ -tame at $s = 1$ with order $k_0 \geq 1$, then the asymptotic order of the mean number $S(n)$ of symbol comparisons performed by each of the three algorithms is described in the third column of Figure 6.2.*

Algorithms	Asymptotic order of $S(n)$ when Λ is tame at $s = s_0$, ($s_0 \in]1, 2[$) with order 1	Asymptotic order of $S(n)$ when Λ is tame at $s = 1$ with order k_0
QuickSort	n^{s_0}	$n \log^{1+k_0} n$
InsSort	n^2	n^2
BubSort	n^{1+s_0}	$n^2 \log^{k_0} n$

Figure 6.2: Results for Proposition 6.4.

This result applies to intermittent sources with parameter a defined in Section 2.2.3. More precisely, Assertion (i) applies for $a \in]1, 1/2[$, and Assertion (ii) applies for $a = 1$, with $k_0 = 2$.

¹The constant κ_2 is not described here.

6.1.5 Robustness.

We now compare the asymptotic estimates for the two mean numbers, the mean number $K(n)$ of key-comparisons (column 2 of Figure 6.1) and the mean number $S(n)$ (column 3 of Figure 6.1). There are two types of algorithms

- (a) The “robust” algorithms for which $K(n)$ and $S(n)$ are of the same order. This is the case for only one algorithm, the **InsSort** algorithm, and the ratio $S(n)/K(n)$ involves the coincidence $c(\mathcal{S})$, described in Section 6.1.3.
- (b) The algorithms for which $S(n)$ and $K(n)$ are not of the same order, here **QuickSort** and **BubSort**. In both cases, the ratio $S(n)/K(n)$ satisfies

$$\frac{S(n)}{K(n)} \sim \frac{1}{2h(\mathcal{S})} \log n. \quad (6.2)$$

We will see later in Section 6.1.6 that the same ratio also appears in lower bounds.

6.1.6 An asymptotic lower bound for $S(n)$.

Combining our methods described in Proposition 6.11 together with the approach of Seidel for lower bounds, we obtain an asymptotic lower bound $\underline{S}(n)$ for the mean number of symbol comparisons for any sorting algorithm (not necessarily faithful) using the standard string comparison procedure and dealing with words of a Λ -tame source \mathcal{S} .

Theorem 6.5. *For a Λ -tame source \mathcal{S} , the following asymptotic lower bound $\underline{S}(n)$ holds for the mean number of symbol comparisons performed by any key-comparison based sorting algorithm and dealing with words emitted by \mathcal{S} ,*

$$\underline{S}(n) \sim \frac{1}{2 \log 2} \frac{1}{h(\mathcal{S})} n \log^2 n.$$

Remark. This lower bound shows that **QuickSort** is quasi-optimal in the model of symbol-comparisons, as it is quasi-optimal in the model of key-comparisons.

6.2 Mean number of symbol comparisons.

We prove in this section the results given in Theorem 6.2 with the help of the framework described in Chapter 5.

6.2.1 Expression for $\pi_n(i, j)$

Figure 6.3 (above) recalls, for each algorithm of interest, the expressions for the mean number $\pi_n(i, j)$ of key comparisons between the key of rank i and the key of rank j . Figure 6.3 (below) considers only the normalized comparisons between keys (see Section 5.2.1 and particularly Eq. (5.2)).

6.2.2 Expression of $\hat{\pi}[u, t]$ in the basis G_m .

We deal with the random variables defined in (5.8) and an easy computation leads to the proof of MetaTheorem 5.4.

The random variable $\hat{\pi}[u, t]$ is expressed in the “basis” $G_m(N_{\mathcal{J}})$, as displayed in the table in Figure 6.4.

Algorithms	$\pi(i, j)$
QuickSort	$\frac{2}{j-i+1}$
InsSort	$\frac{1}{2} + \frac{1}{(j-i+1)(j-i)}$
BubSort	$\frac{1}{2} + \frac{1}{(j-i+1)(j-i)} + \frac{2(i-1)}{(j-i+2)(j-i+1)(j-i)}$

Algorithms	$\hat{\pi}_n(i, j)$
QuickSort	$\frac{2}{j-i+1}$
InsSort	$\frac{1}{(j-i+1)(j-i)}$
BubSort	$\frac{1}{(j-i+1)(j-i)} + \frac{2(i-1)}{(j-i+2)(j-i+1)(j-i)}$

Figure 6.3: (Above) Expression of the mean number $\pi_n(i, j)$ of key comparisons between the key of rank i and the key of rank j for each sorting algorithm. (Below) Expression of the mean number $\hat{\pi}_n(i, j)$ of normalized key comparisons (i.e., omitting the constant $1/2$).

6.2.3 Expression of the coefficient $\varphi(k, u, t)$

We obtain an explicit expression for the k -th coefficient $\varphi(k, u, t)$ in the series expansion as in (5.7),

$$\Phi_Z(u, t) = \sum_{k \geq 1 + \sigma_0} \frac{Z^k}{k!} (-1)^k \varphi(k, u, t)$$

It is deduced from the decomposition described in Figure 6.4 together with expressions in (5.9). This leads to the proof of MetaTheorem 5.6. We recall that $\sigma_0 + 1$ is (by definition) the first integer where the decomposition is valid.

For each sorting algorithm, the value of the integer σ_0 is provided in the second column of Figure 6.5 and the expression of $\varphi(k, u, t)$ is stated in the third and fourth column of Figure 6.5.

6.2.4 Expressions of the mixed Dirichlet series $\varpi(s)$.

With computations of the previous expressions over the fundamental triangles, we obtain the expression of $\varphi(k)$, then the analytic lifting $\varpi(s)$ (on the halfplane $\Re s > \sigma_0$) which is also the mixed Dirichlet series:

Consider any general source, together with the fundamental intervals $[a_w, b_w]$ and its Dirichlet series $\Lambda(s)$. For any of the three algorithms, the mixed Dirichlet series $\varpi(s)$ admits in the domain $\Re s > \sigma_0$, the expressions displayed in the second column of Table of Figure 6.6, together with the values of σ_0 in the third column, and the main term in the singular expression of $\varpi(s)/(s - \sigma_0)$.

Algorithms	$\hat{\pi}[u, t]$ (in the “basis” G_m)
QuickSort	$2[G_1(N_{[u,t]}) - G_2(N_{[u,t]})]$
InsSort	$G_2(N_{[u,t]})$
BubSort	$G_2(N_{[u,t]}) + 2N_{[0,u[} \cdot G_3(N_{[u,t]})$

Figure 6.4: Decomposition of $\hat{\pi}_N(u, t)$ in the basis G_m

Algorithms	σ_0	$\varphi(k, u, t), k \geq 1 + \sigma_0$	$\varphi(k, u, t), k \geq 1 + \sigma_0$
QuickSort	1	$2[\beta_1(k, t - u) - \beta_2(k, t - u)]$	$2(t - u)^{k-2}$
InsSort	1	$\beta_2(k, t - u)$	$(k - 1)(t - u)^{k-2}$
BubSort	2	$\beta_2(k, t - u) + 2u\gamma_3(k, t - u)$	$(k - 1)(t - u)^{k-3}[t - (k - 1)u]$

Figure 6.5: Expressions for $\varphi(k, u, t)$.

6.2.5 Tameness of the mixed Dirichlet series.

In the three cases, the mixed Dirichlet series is closely related to the Dirichlet series $\Lambda(s)$ of the source, and the transfer of tameness between $\Lambda(s)$ and $\varpi(s)$ is easy.

Proposition 6.6. *Assume the source \mathcal{S} to be Λ -tame. Then, the mixed Dirichlet series $\varpi(s)$ relative to sorting algorithms satisfy the following:*

- [QuickSort] $\varpi(s)$ is tame at $\sigma_0 = 1$ with order $k_0 = 2$.
- [InsSort] $\varpi(s)$ is tame at $\sigma_0 = 1$ with order $k_0 = 1$.
- [BubSort] $\varpi(s)$ is tame at $\sigma_0 = 2$ with order $k_0 = 1$.

Moreover, the source gives its shape of tameness to the mixed Dirichlet series.

Proof. The cases of QuickSort and InsSort is different from the one of BubSort.

Case of QuickSort and InsSort. The integral of $(s - 1)(t - u)^{s-2}$ on the fundamental triangle \mathcal{T}_w equals $(1/s)p_w^s$. This entails the nice formula for both $\varpi_Q(s)$ and $\varpi_I(s)$ (where Q and I stand respectively for QuickSort and InsSort)

$$\varpi_Q(s) = \frac{\Lambda(s)}{s(s - 1)}, \quad \varpi_I(s) = \frac{\Lambda(s)}{s}.$$

Then, the functions $s \mapsto \varpi(s)$ are tame at $s = 1$. Moreover, the shape of tameness of $\varpi(s)$ at $s = 1$ coincides with the shape of Λ -tameness of the source. For InsSort, the function $\varpi_I(s)$ has a simple dominant pole at $s = 1$ with a residue equal to $1/h(S)$, whereas, for QuickSort, the function $\varpi_I(s)$ has a dominant pole at $s = 1$ of order 2. Moreover, the singular expressions of the functions $\varpi(s)/(s - 1)$ can be easily computed from the singular expression of $\Lambda(s)$.

Algorithms	$\varpi(s)$	$\varpi(s)$	σ_0	Main term of $\varpi(s)/(s - \sigma_0)$
QuickSort	$\frac{2}{s(s-1)} \sum_{w \in \Sigma^*} p_w^s$	$\frac{2}{s(s-1)} \Lambda(s)$	1	$\frac{2}{h(\mathcal{S})} \frac{1}{(s-1)^3}$
InsSort	$\frac{1}{s} \sum_{w \in \Sigma^*} p_w^s$	$\frac{1}{s} \Lambda(s)$	1	$\frac{1}{h(\mathcal{S})} \frac{1}{(s-1)^2}$
BubSort	$-\sum_{w \in \Sigma^*} a_w p_w^{s-1}$	$-\Lambda[F_0](s-1)$	2	$-\frac{1}{2h(\mathcal{S})} \frac{1}{(s-2)^2}$

Figure 6.6: Expressions for $\varpi(s)$. (N.B.: The function F_0 is $F_0(x, y) = x$.)

Case of BubSort. The integral of $\varpi(s, u, t) = (s-1)(t-u)^{s-3}[t - (s-1)u]$ over the fundamental triangle equals $-a_w p_w^{s-1}$. Then, the Dirichlet series $\varpi(s)$ admits the expression

$$\varpi(s) = - \sum_{w \in \Sigma^*} a_w p_w^{s-1} = -\Lambda[F_0](s-1),$$

where $F_0(x, y) = x$. By hypothesis, the series $s \mapsto \Lambda[F_0](s)$ is tame at $s = 1$. Then, the series $\varpi(s)$ is tame at $s = 2$, with the same shape of tameness as the series $s \mapsto \Lambda[F_0](s)$. We now study its precise behavior at $s = 2$. We remark, with the relation $\Lambda_\ell(1) = \Lambda_\ell(1)^2 = 1$, the equality

$$2\Lambda_\ell[F_0](1) = 2 \sum_{w \in \Sigma^\ell} a_w p_w = 2 \sum_{w \in \Sigma^\ell} \left[\sum_{w' < w} p_{w'} \right] p_w = \Lambda_\ell(1)^2 - \sum_{w \in \Sigma^\ell} p_w^2 = \Lambda_\ell(1) - \Lambda_\ell(2). \quad (6.3)$$

The series

$$L(s) := \sum_{\ell \geq 0} L_\ell(s) \quad \text{with} \quad L_\ell(s) := 2\Lambda_\ell[F_0](s) - \Lambda_\ell(s),$$

is convergent at $s = 1$ and satisfies

$$L(1) = \sum_{\ell \geq 0} L_\ell(1) = - \sum_{\ell \geq 0} \Lambda_\ell(2) = -\Lambda(2) = -c(\mathcal{S}),$$

where $c(\mathcal{S})$ is the coincidence of the source defined in Lemma 2.5. Since $\Lambda(s)$ admits a simple pole at $s = 1$ with a residue equal to $1/h(\mathcal{S})$, then $\Lambda[F_0](s)$ admits a simple pole at $s = 1$ with a residue equal to $1/2h(\mathcal{S})$. More precisely, given the singular expansion of $\Lambda(s)$ at $s = 1$ written as

$$\Lambda(s) = \frac{1}{h(\mathcal{S})} \frac{1}{s-1} + d(\mathcal{S}) + O(s-1),$$

the singular expansion of $\varpi(s)$ at $s = 2$ is

$$\varpi(s) = -\frac{1}{2h(\mathcal{S})} \frac{1}{s-2} + \frac{1}{2}(c(\mathcal{S}) - d(\mathcal{S})) + O(s-2).$$

Remark that Equation (6.3) can be generalized to any function F of class \mathcal{C}^1 . The sum of interest can be viewed as a Riemann sum on the fundamental intervals of depth ℓ , so that the decomposition holds

$$\sum_{w \in \Sigma^\ell} F(a_w, b_w) p_w = I[F] + \rho_\ell[F].$$

where the main term is the integral $I[F]$ of the diagonal function $x \mapsto F(x, x)$ and the remainder term $\rho_\ell[F]$ is the difference between the integral and the Riemann sum, that satisfies

$$|\rho_\ell[F]| \leq \sum_{w \in \Sigma^\ell} \int_{a_w}^{b_w} |F(a_w, b_w) - F(x, x)| dx.$$

On each interval $[a_w, b_w]$, the inequality holds (with D the differential),

$$|F(x, x) - F(a_w, b_w)| \leq C \cdot |a_w - b_w|, \quad \text{with } C := \sup_{(x,y) \in [0,1]^2} |DF(x, y)|,$$

and entails the bound $\rho_\ell[F] \leq C \cdot \Lambda_\ell(2)$. Then, for any function $F \geq 0$ of class \mathcal{C}^1 , with $F \not\equiv 0$, the integral $I[F]$ is not zero, and the Dirichlet series $\Lambda[F](s)$ has a residue at $s = 1$ equal to $I[F]/h(\mathcal{S})$. \square

The singular expressions obtained when studying the tameness of $\varpi(s)$ are stated in the last column of the table in Figure 6.6. It is then a simple matter to apply the Rice method to obtain (by residue calculus) the asymptotic expansions needed in Theorem 6.2 (or in the last column of Figure 6.1).

6.2.6 Mean number of symbol comparisons

One last step in order to get the last column of Figure 6.1 is to take into account the non-normalized comparisons. The mean number $S(n)$ of symbol comparisons is

$$S(n) = \begin{cases} \widehat{S}(n) & \text{(for QuickSort),} \\ \widehat{S}(n) + \binom{n}{2} \frac{\Lambda(2)}{2} & \text{(for InsSort and BubSort).} \end{cases}$$

Remark that the extra term which is added is the main term for **InsSort**, for which the cost of non normalized comparisons is the leading one.

6.3 An alternative approach.

We first recall the approach due to Seidel [65], together with the notion of faithfulness he introduced (Section 6.3.3). This approach leads to analyze additive trie parameters (Section 6.3.4). We then make a “detour” via tries and trie parameters in Section 6.3.5, and we use the methods of Chapter 4 in such analyses: the Rice method and the Poisson-Mellin approach. When the first one may be applied (this is the case for our analyses in previous sections), this greatly simplifies the probabilistic analysis. But, we also explain how to deal with the second method when the first one cannot be applied. Then, in Section 6.3.7, we return to faithful sorting algorithms and describe how to mix the two approaches (Seidel’s one and the analysis of Section 6.3.5). This leads to an alternative derivation for results for **QuickSort** and **InsSort** algorithms.

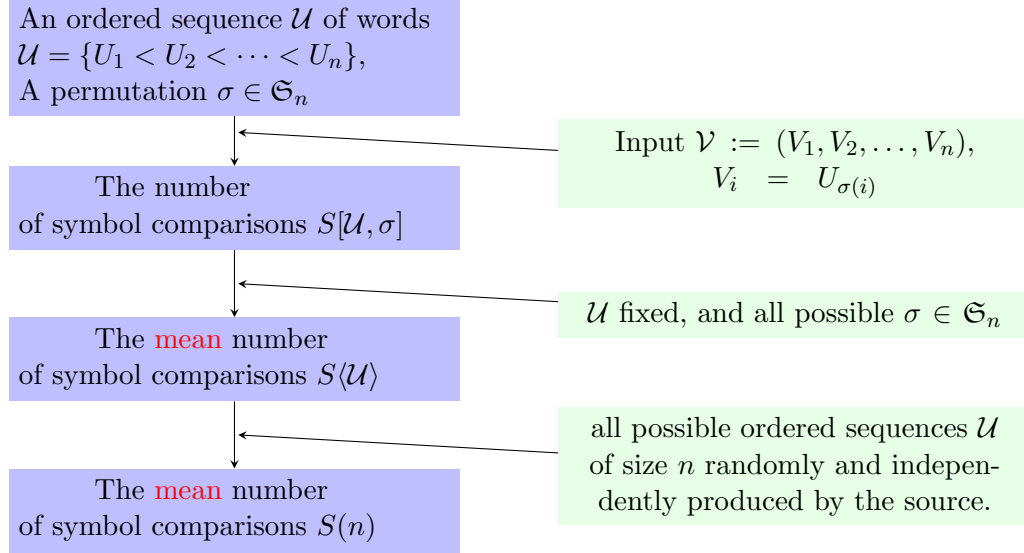


Figure 6.7: The probabilistic steps for the analysis.

6.3.1 Seidel's point of view.

Seidel [65] introduces the following framework, which is well-described in Figure 6.7.

Definition 6.7. Consider a set \mathcal{U} of words that are independently emitted by the same source \mathcal{S} . Denote by $S[\mathcal{U}, \sigma]$ the number of symbol (resp. $K[\mathcal{U}, \sigma]$ the number of key comparisons performed by the algorithm when the input set \mathcal{U} is under the permutation σ . More generally, for a subset $\mathcal{V} \subset \mathcal{U}$, denote by $S[\mathcal{U}, \mathcal{V}, \sigma]$ (resp. $K[\mathcal{U}, \mathcal{V}, \sigma]$) the number of symbol (resp. key) comparisons performed by the algorithm on the subset \mathcal{V} when the input set \mathcal{U} is under the permutation σ . The quantities $S\langle \mathcal{U} \rangle$, $S\langle \mathcal{U}, \mathcal{V} \rangle$, (resp. $K\langle \mathcal{U} \rangle$, $K\langle \mathcal{U}, \mathcal{V} \rangle$) are the expected values of $S[\mathcal{U}, \sigma]$, $S[\mathcal{U}, \mathcal{V}, \sigma]$ (resp. $K[\mathcal{U}, \sigma]$, $K[\mathcal{U}, \mathcal{V}, \sigma]$) when σ is a uniform random permutation of \mathfrak{S}_n .

Remark The notation $\langle \cdot \rangle$ denotes an average over all the possible sequences that “come from” the same set.

6.3.2 Seidel's result

Seidel [65] proves the following result that is now described in our framework.

Proposition 6.8 (Seidel, 2010). Consider the previous framework. Denote by $\mathcal{U}_{\langle w \rangle}$ the subset of \mathcal{U} which gathers all the words which begin by the prefix w , and by $P(\mathcal{U})$ the set of common prefixes of \mathcal{U} , defined as the prefixes w for which the cardinality $|\mathcal{U}_{\langle w \rangle}|$ is at least equal to 2. The following relations hold:

$$S[\mathcal{U}, \sigma] = \sum_{w \in P(\mathcal{U})} K[\mathcal{U}, \mathcal{U}_{\langle w \rangle}, \sigma], \quad S\langle \mathcal{U} \rangle = \sum_{w \in P(\mathcal{U})} K\langle \mathcal{U}, \mathcal{U}_{\langle w \rangle} \rangle. \quad (6.4)$$

Proof. The following equality

$$S[\mathcal{U}, \sigma] = \sum_{1 \leq i < j \leq n} (c(U_i, U_j) + 1) K[\mathcal{U}, \{U_i, U_j\}, \sigma],$$

involves (by definition) the coincidence $c(U_i, U_j)$ between the two keys U_i and U_j (namely the length of their longest common prefix). The set $P(\mathcal{U})$ of common prefixes of \mathcal{U} exactly gathers the prefixes w for which the cardinality $|\mathcal{U}_{\langle w \rangle}|$ is at least equal to 2. All these sets only depend on \mathcal{U} , not on the permutation σ .

Seidel remarks that $c(U_i, U_j) + 1$ is also the number of prefixes $w \in P(\mathcal{U})$ —including the empty prefix— which are shared by U_i and U_j . This is also the number of subsets of the form $\mathcal{U}_{\langle w \rangle}$ — including the total set \mathcal{U} — which contain U_i and U_j . Then,

$$S[\mathcal{U}, \sigma] = \sum_{1 \leq i < j \leq n} \sum_{\substack{w \in P(\mathcal{U}) \\ U_i, U_j \in \mathcal{U}_{\langle w \rangle}}} K[\mathcal{U}, \{U_i, U_j\}, \sigma] = \sum_{w \in P(\mathcal{U})} \sum_{\substack{1 \leq i < j \leq n \\ U_i, U_j \in \mathcal{U}_{\langle w \rangle}}} K[\mathcal{U}, \{U_i, U_j\}, \sigma].$$

Now the equality

$$\sum_{\substack{1 \leq i < j \leq n \\ U_i, U_j \in \mathcal{U}_{\langle w \rangle}}} K[\mathcal{U}, \{U_i, U_j\}, \sigma] = K[\mathcal{U}, \mathcal{U}_{\langle w \rangle}, \sigma],$$

holds, and entails the first equality of Proposition 6.8. As the subsets $\mathcal{U}_{\langle w \rangle}$ do not depend on the permutation σ , averaging over σ leads to the second equality of Proposition 6.8. \square

6.3.3 Faithfulness.

Seidel introduces the notion of a faithful algorithm, and we consider here a slightly different notion, the notion of a strongly faithful algorithm, on which we give an alternative point of view. We first recall the notation $\mathcal{U}_{[i,j]}$ (already used in Section 1.1.4) which denotes the subset formed with the keys of \mathcal{U} whose rank k belongs to the interval $[i, j]$.

Definition 6.9 (Seidel, 2010). *An algorithm is strongly faithful if, for any $n \geq 2$, any subset \mathcal{U} of cardinality n , and any pair (i, j) , with $1 \leq i < j \leq n$, the mean number of key comparisons $\mathcal{K}(\mathcal{U}, \mathcal{U}_{[i,j]})$ performed by the algorithm only depends on the cardinality $j - i + 1$ of the subset $\mathcal{U}_{[i,j]}$.*

There is an easy translation of this notion in our framework.

Lemma 6.10. *For a sorting algorithm \mathcal{A} , the following three assertions are equivalent*

- (a) *The algorithm is strongly faithful.*
- (b) *The mean number of key comparisons $\pi(i, j)$ performed by \mathcal{A} between the two keys U_i and U_j only depends on the difference $j - i$ between their ranks.*
- (c) *The density $\Phi_Z(u, t)$ of \mathcal{A} in the Poisson model only depends on the difference $t - u$.*

*The algorithms **Quicksort** and **InsSort** are strongly faithful. The third algorithm **BubSort** is not strongly faithful.*

Proof. Denote by $P(i, j) := \mathcal{K}[\mathcal{U}, \mathcal{U}_{[i,j]}]$ the expectation of the total number of key comparisons between any pair of two keys of $\mathcal{U}_{[i,j]}$. By definition, an algorithm is strongly faithful if $P(i, i + k)$ does not depend on i , and only depends on k .

The relation

$$\begin{aligned} P(i, i + \ell) &= \sum_{\substack{(i', j') \\ i \leq i' < j' \leq i + \ell}} \pi(i', j') \\ &= P(i, i + \ell - 1) + \pi(i, i + \ell) + \sum_{k=1}^{\ell-1} \pi(i + k, i + \ell), \end{aligned} \tag{6.5}$$

holds for $\ell \geq 2$ between the two sequences $P(i, j)$ and $\pi(i, j)$. We will use it to prove that the following two assertions are equivalent, with a recurrence on ℓ .

- (a) For any $k \leq \ell$, the expected values $P(i, i + k)$ do not depend on i .
- (b) For any $k \leq \ell$, the expected values $\pi(i, i + k)$ do not depend on i .

If $\ell = 1$, there is only one pair of keys in $\mathcal{U}_{[i, i+1]}$ and $P(i, i + 1) = \pi(i, i + 1)$. Then, the lemma is true for $\ell = 1$.

Let us suppose now that the lemma holds for $k < \ell$. We will prove that it holds for $k = \ell$.

Assume first that Assertion (b) holds for $k \leq \ell$. Then, none of the terms $\pi(i + k, i + \ell)$ for $k \in [0 \dots \ell - 1]$ depends on i . Furthermore, by recurrence hypothesis, Assertion (a) holds for $k \leq \ell - 1$, and $P(i, i + \ell - 1)$ does not depend on i . Then, with Eq. (6.5), it is the same for $P(i, i + \ell)$. and Assertion (a) holds for $k \leq \ell$.

Conversely, assume that Assertion (a) holds for $k \leq \ell$. Then, none of the two terms $P(i, i + \ell)$ or $P(i, i + \ell - 1)$ depends on i . Furthermore, by recurrence hypothesis, Assertion (b) holds for $k < \ell$ and all the terms $\pi(i + k, i + \ell)$ are independent of i for $k \in [1 \dots \ell]$. Then, with Eq. (6.5), it is the same for $\pi(i, i + \ell)$, and Assertion (b) holds for $k \leq \ell$. \square

For a strongly faithful algorithm, the mean number $K\langle \mathcal{U}, \mathcal{U}_{\langle w \rangle} \rangle$ only depends on the cardinality N_w of $\mathcal{U}_{\langle w \rangle}$. It equals $K(N_w)$, where $K(n)$ is the mean number of key comparisons of the algorithm in the permutation model, and Relation (6.4) entails the equality

$$S\langle \mathcal{U} \rangle = \sum_{w \in \mathcal{P}(\mathcal{U})} K(N_w), \quad (6.6)$$

where N_w is the number of words of \mathcal{U} which begin with the prefix w . We remark that, for any sorting algorithm, the equalities $K(0) = K(1) = 0$ hold. Then, the previous relation can be written as

$$S\langle \mathcal{U} \rangle = \sum_{w \in \mathcal{P}(\mathcal{U})} K(N_w) = \sum_{w \in \Sigma^*} K(N_w).$$

6.3.4 A relation between faithful sorting algorithms and trie parameters.

Consider, more generally, a function $f : \mathbb{N} \rightarrow \mathbb{R}$ which satisfies $f(0) = f(1) = 0$ and $f(k) \geq 0$ for $k \geq 2$, and a random variable R defined on \mathcal{M}^* by the relation

$$R[\mathcal{U}] := \sum_{w \in \Sigma^*} f(N_w), \quad (6.7)$$

where N_w is the number² of words of \mathcal{U} which begin with the prefix w . Remark that the right member only depends on the underlying set, then the equality $R[\mathcal{U}] = R\langle \mathcal{U} \rangle$ holds. We now explain why such a random variable defines an additive parameter on the trie $T(\mathcal{U})$.

²This notation is in conflict with the notations used in the other chapters. Previously in the thesis, we have used $N_{\mathcal{J}}$ to denote the number of words whose *parameter* belongs to \mathcal{J} . Then we should use $N_{\mathcal{I}_w}$ instead of N_w .

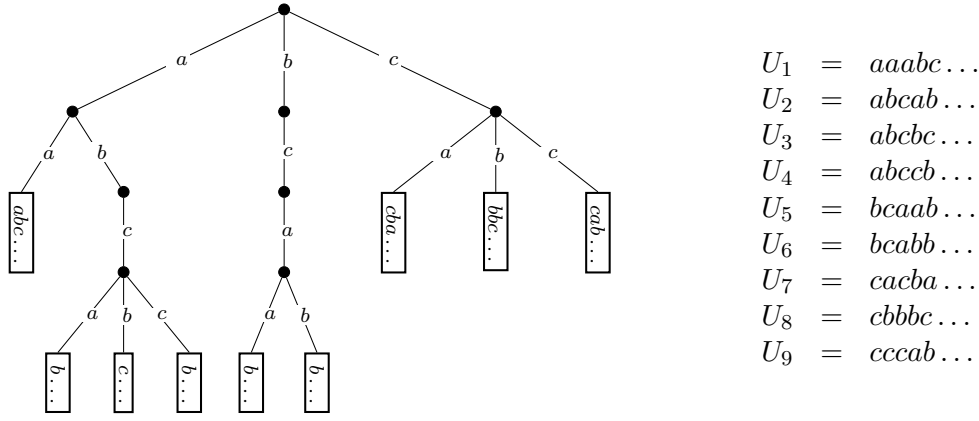


Figure 6.8: The trie $T(\mathcal{U})$ associated with a set \mathcal{U} of nine (infinite) words on the alphabet $\Sigma := \{a, b, c\}$

The trie $T(\mathcal{U})$. A trie is a tree structure, used as a dictionary, which compares words via their prefixes. Since its introduction [31, 19] the trie has become a fundamental data structure in computer science [24]. Given a finite set $\mathcal{U} = \{U_1, U_2, \dots, U_n\}$ formed with n (infinite) words emitted by the source, the trie $T(\mathcal{U})$ built on the set \mathcal{U} is defined recursively by the following three rules:

- (i) If $|\mathcal{U}| = 0$, $T(\mathcal{U}) = \emptyset$.
- (ii) If $|\mathcal{U}| = 1$, $\mathcal{U} = \{U\}$, $T(\mathcal{U})$ is a leaf labeled by U .
- (iii) If $|\mathcal{U}| \geq 2$, then $T(\mathcal{U})$ is formed with an internal node and r subtrees respectively equal to

$$T(\underline{\mathcal{U}}_{(0)}), \dots, T(\underline{\mathcal{U}}_{(r-1)}),$$

where $\underline{\mathcal{U}}_{(\sigma)}$ denotes the subset consisting of words of $\mathcal{U}_{(\sigma)}$ (i.e., words beginning with letter σ), stripped of their initial symbol σ . If the set $\mathcal{U}_{(\sigma)}$ is nonempty, the edge which links the subtree $T(\underline{\mathcal{U}}_{(\sigma)})$ to the internal node is labelled with the symbol σ .

Then, the internal nodes are used for directing the search, and the leaves contain suffixes of words of \mathcal{U} . There are as many leaves as words in \mathcal{U} . Internal nodes are labeled by prefixes w for which the cardinality N_w of the subset $\mathcal{U}_{(w)}$ is at least 2 (see Figure 6.8).

Additive parameters on tries. Trie analysis aims at describing the average shape of a trie (number of internal nodes, external path length, height). We focus here on additive parameters, whose (recursive) definition exactly copies the (recursive) definition of the trie. Consider a function $f : \mathbb{N} \rightarrow \mathbb{R}$ which satisfies $f(0) = f(1) = 0$ and $f(k) \geq 0$ for $k \geq 2$, together with a random variable $R[\mathcal{U}]$, associated with f , and defined on the trie $T(\mathcal{U})$ as:

- (i) If $|\mathcal{U}| < 2$, then $R[\mathcal{U}] = 0$;
- (ii) If $|\mathcal{U}| \geq 2$, then $R[\mathcal{U}] = f(|\mathcal{U}|) + \sum_{\sigma \in \Sigma} R[\mathcal{U}_{(\sigma)}]$.

Iterating the recursion, we obtain exactly Equation (6.7). The cost f is the “toll” that is “paid” at each internal node of the trie. In particular, when $f(k) = 1$ for $k \geq 2$, the quantity $\mathcal{R}[\mathcal{U}]$ equals the number of internal nodes in the trie $T(\mathcal{U})$, and when $f(k) = k$ for $k \geq 2$, the variable $\mathcal{R}[\mathcal{U}]$ equals the external path length of the trie $T(\mathcal{U})$. These trie parameters have been very deeply studied: first in the case when the words are emitted

by a simple source (see [46] for instance), and later on, when the words are produced by a general (non-ambiguous) source (see [13, 11]).

However, even for simple sources, these existing analyses are usually performed with the Poisson-Mellin tools, need Depoissonization techniques, and do not precisely deal with the tameness of the source³. The following result has thus two main purposes: it first deals with the usual cases $f(k) = 1$ or $f(k) = k$ and explains how the Rice method provides in these cases very natural proofs, with precise error terms, that do not need Depoissonization techniques. It also makes more precise the role played by the tameness of the source. Second, it describes the method that can be used in the case of the “toll” $f(k) = k \log k$, where it does not seem possible to apply directly the Rice method.

6.3.5 Analysis of additive trie parameters.

The following result compares the two methodologies, the Rice method and the Poisson-Mellin approach, it is thus of independent interest, and also important in our context.

Proposition 6.11. *Consider, a source \mathcal{S} assumed to be Λ -tame. For each set⁴ \mathcal{U} of infinite words independently produced by \mathcal{S} , consider the trie $T(\mathcal{U})$ and a trie parameter R defined by Relation (6.7) from a toll function $f : \mathbb{N} \rightarrow \mathbb{R}$ which satisfies $f(0) = f(1) = 0$ and $f(k) \geq 0$ for $k \geq 2$. Then, the mean value $R(n)$ of the random variable R in the Bernoulli model $(\mathcal{B}_n, \mathcal{S})$ satisfies the following:*

- (i) *In the case when $f(k) = k(k-1)$, then $R(n) = \Lambda(2)n(n-1)$.*
- (ii) *Define the degree of f as $\deg(f) := \inf\{c, f(k) = O(k^c)\}$ and assume that $d := \deg f$ belongs to $[0, 2[$. Then, $R(n)$ is written as*

$$R(n) = \sum_{k=2}^n (-1)^k \binom{n}{k} \Lambda(k) r(k), \quad \text{with} \quad r(n) = \sum_{k=2}^n (-1)^k \binom{n}{k} f(k),$$

and there exist analytic liftings $\rho(s)$, for $n \mapsto r(n)$, and $\varpi(s)$, for $n \mapsto R(n)$, on the half-plane $\Re s > d$.

- (iii) *In the case when $\rho(s)$ is proven to be tame at $s = \max(d, 1)$, the Rice method can be applied. This arises in particular in the two cases $f(k) = 1$ and $f(k) = k$. The function ρ , and the asymptotic behavior of the mean value R are described in Figure 6.9.*
- (iv) *In the case when $f(k) = k \log k$, the function $\rho(s)$ has a pole of order 2 at $s = 1$, and, even though it is not proven to be tame at $s = 1$, Poisson-Mellin tools prove the asymptotic behaviour for $R(n)$ given in Figure 6.9.*

Remark. There exist two extensions of Proposition 6.11 (that will be described in [69]), in the following cases:

- (i) The source is not Λ -tame, but its Λ series fulfills nice tameness properties as described in Section 6.1.4.
- (ii) The function f admits an analytic lifting that satisfies $f(n) = n^a \log^b n$, with $b \in \mathbb{N}$ and $a \in]0, 2[$.

³There is an exception in [11].

⁴Or sequence since it does not matter for tries.

Toll function	Lifting $\rho(s)$	Lifting $\varpi(s)$	Dominant term of $R(n)$
$f(k) = 1$	$s - 1$	$(s - 1)\Lambda(s)$	$\frac{n}{h(\mathcal{S})}$
$f(k) = k$	s	$s\Lambda(s)$	$\frac{1}{h(\mathcal{S})}n \log n$
$f(k) = k \log k$	$-\frac{\zeta'(s)}{\Gamma(-s)} + \frac{H_1(s)}{\Gamma(-s)}$	$-\frac{\zeta'(s)\Lambda(s)}{\Gamma(-s)} + \Lambda(s)\frac{H_1(s)}{\Gamma(-s)}$	$\frac{1}{2h(\mathcal{S})}n \log^2 n$

Figure 6.9: Results for Proposition 6.11. The function $H_1(s)$ is defined in (6.13).

6.3.6 Proof of Proposition 6.11.

We first work inside the Poisson model $(\mathcal{P}_Z, \mathcal{S})$ where the cardinality N follows a Poisson law of rate Z . Then, the cardinality N_w follows a Poisson law of rate Zp_w . Denote by $F(Z)$ the expectation of the variable $f(N)$ and $G(Z)$ the expectation of the variable $\mathcal{R}[\mathcal{U}]$ in the Poisson model of rate Z , namely

$$F(Z) = e^{-Z} \sum_{k \geq 2} \frac{Z^k}{k!} f(k), \quad G(Z) = e^{-Z} \sum_{k \geq 2} \frac{Z^k}{k!} R(k), \quad (6.8)$$

where $R(n)$ is the expectation of $\mathcal{R}[\mathcal{U}]$ in the Bernoulli model in $(\mathcal{B}_n, \mathcal{S})$. If we wish to adopt the Rice method, as in Chapter 4, we write $F(Z), G(Z)$ under the form

$$F(Z) = \sum_{n \geq 0} (-1)^n \frac{Z^n}{n!} r(n), \quad \text{with} \quad r(n) = \sum_{k=2}^n (-1)^k \binom{n}{k} f(k), \quad (6.9)$$

$$G(Z) = \sum_{n \geq 0} (-1)^n \frac{Z^n}{n!} \varphi(n), \quad \text{with} \quad \varphi(n) = \sum_{k=2}^n (-1)^k \binom{n}{k} R(k). \quad (6.10)$$

Averaging Relation (6.7) in the Poisson model of rate Z entails the equality

$$G(Z) = \sum_{w \in \Sigma^*} \mathbb{E}_Z[f(N_w)] = \sum_{w \in \Sigma^*} F(Zp_w). \quad (6.11)$$

Assertion (i). Consider first the case where $f(k) = k(k-1)$. Then, Relation (6.8) entails the equality $F(Z) = Z^2$, and, with (6.11), the equality $G(Z) = \Lambda(2)Z^2$. This implies the *exact* equality $R(n) = n(n-1)\Lambda(2)$.

Assertion (ii). Relations (6.9), (6.10) and (6.11) entail the equality

$$\varphi(n) = \left(\sum_{w \in \Sigma^*} p_w^n \right) r(n) = \Lambda(n)r(n), \quad (6.12)$$

and, inverting the triangular set of relations (6.10) leads to the binomial relations

$$R(n) = \sum_{k=2}^n (-1)^k \binom{n}{k} \varphi(k), \quad \text{and then} \quad R(n) = \sum_{k=2}^n (-1)^k \binom{n}{k} \Lambda(k)r(k).$$

If there exists an analytic lifting $\rho(s)$ of $r(n)$, there is an analytic lifting $\varpi(s)$ for $\varphi(n)$, equal to $\Lambda(s)\rho(s)$, and we can use the Rice Formula, as previously in Section 6.2.5, as soon

as the analytic liftings can be proven tame. Previously, for each of the three algorithms, we computed such an analytic lifting $\varpi(s)$, given in the second column of Table of Figure 6.6, which is moreover proven to be *tame*. However, in the present general setting, the existence of a simple lifting $\varpi(s)$ for $\varphi(k)$ is not always directly granted, but it can be obtained by the cycle “Poisson–Mellin–Newton–Rice”, that we have described in Section 4.6.

First, the Valuation-Degree Condition is satisfied for the sequence $f(k) = k \log k$ that satisfies $\deg f = d \in [0, 2[$ and $\text{val}(f) = 2$. Then, we use Proposition 4.16 that proves the existence of an analytic lifting $\rho(s)$ for the sequence $r(k)$ on the half plane $\Re s > 1$. As the Dirichlet series Λ of the source has a pole at $s = 1$, the product $s \mapsto \varpi(s) = \rho(s)\Lambda(s)$ is analytic on the halfplane $\Re s > \max(d, 1)$. And its tameness (needed to apply the Rice formula) depends both on the relative position of d and 1, and the tameness of ρ .

We now describe particular instances of such a situation, in the context of Assertions (iii) and (iv).

Assertion (iii). We now consider the cases $f(k) = 1$ or $f(k) = k$. In both cases, the degree d satisfies $\max(d, 1) = 1$, and we study the tameness of ρ at $s = 1$. The sequences $r(n)$ satisfy respectively

$$\begin{aligned} r(n) &= \sum_{k=2}^n (-1)^k \binom{n}{k} = 0 - 1 + n = n - 1, \\ r(n) &= \sum_{k=2}^n (-1)^k \binom{n}{k} k = -n \sum_{k=1}^{n-1} (-1)^k \binom{n-1}{k} = n, \end{aligned}$$

and the analytic liftings $\rho(s)$ are respectively $s-1$ and s , whereas the analytic liftings of the sequence φ are $\varpi(s) = (s-1)\Lambda(s)$ and $s\Lambda(s)$. In the case of a source Λ -tame, the analytic lifting $\varpi(s)$ is tame at $s = 1$, and we apply the Rice method. We then easily recover the main results already known for the asymptotic mean of the size and path length, with precise remainder terms which depend on the tameness of the source.

Assertion (iv). The case $f(k) = k \log k$ is different. The degree d equals 1, and we study the tameness of ρ at $s = 1$. We write

$$F^*(s) = \sum_{k \geq 2} \frac{f(k)}{k!} \int_0^\infty e^{-z} z^k z^{s-1} dz = \sum_{k \geq 2} \frac{f(k)}{k} \frac{\Gamma(k+s)}{\Gamma(k)}.$$

The ratio of Gamma Functions can be estimated with the Stirling Formula,

$$\frac{\Gamma(k+s)}{\Gamma(k)} = \frac{(k+s)^{k+s}}{k^k} \frac{e^{-k-s}}{e^{-k}} \sqrt{\frac{k+s}{k}} \left[1 + O\left(\frac{1}{k}\right) \right] = k^s \left[1 + O\left(\frac{|s|}{k}\right) \right],$$

where the O -term is uniform with respect to k . Then, the Mellin transform of F satisfies, for $f(k) = k \log k$,

$$F^*(s) = \sum_{k \geq 2} k^s \log k \left[1 + O\left(\frac{|s|}{k}\right) \right] = -\zeta'(-s) + H_1(s), \quad (6.13)$$

where $H_1(s)$ is analytic and of polynomial growth in $\Re s < 0$. Then $F^*(-s)$ is tame at $s = 1$ with order 2, and the relation

$$\rho(s) = \frac{F^*(-s)}{\Gamma(-s)} = -\frac{\zeta'(s)}{\Gamma(-s)} + \frac{H_1(s)}{\Gamma(-s)},$$

proves that the function $\rho(s)$ has a pole of order 1 at $s = 1$. However, the function $1/\Gamma(-s)$, even though it is analytic on the halfplane $\Re(s) > 1$, is not tame there. And it does not seem possible to directly prove that $\rho(s)$ is tame at $s = 1$. We cannot *a priori* apply the Rice method, and we thus follow the alternative approach, using the Mellin-Poisson transform, followed by Depoissonization. These techniques not yet used in this thesis, are described in Chapter 4.2.

We also use some easy facts on the Mellin transform, described in Section 4.5. We first consider the Mellin transforms $F^*(s)$ and $G^*(s)$. Due to Relation (6.11), the function $G(Z)$ is an harmonic sum, with base function F and frequencies p_w (see Section 4.5 and [25]). With classical properties recalled in Section 4.5.2, its Mellin transform $G^*(s)$ factorizes as

$$G^*(s) = \left(\sum_{w \in \Sigma^*} p_w^{-s} \right) \cdot F^*(s) = \Lambda(-s) \cdot F^*(s). \quad (6.14)$$

The singular expressions of $F^*(s)$ and $G^*(s)$ at $s = -1$ are, with tameness of $\Lambda(s)$ and (6.14),

$$F^*(s) \asymp \frac{1}{(s+1)^2}, \quad G^*(s) \asymp \frac{1}{h(\mathcal{S})} \frac{1}{(s+1)^3}.$$

The tamenesses of $F^*(s)$ and $\Lambda(s)$ are enough to deduce, using standard Mellin inverse transform, recalled in Section 4.5, the estimates, for $Z \rightarrow \infty$,

$$F(Z) = Z \log Z (1 + o(1)), \quad G(Z) = \frac{1}{2h(\mathcal{S})} Z \log^2 Z (1 + o(1)). \quad (6.15)$$

Now, we wish to return to the Bernoulli model, with Depoissonization techniques, which need a good behaviour of $G(Z)$ with respect to cones. We use the theorem due to Jacquet and Szpankowski, described in Section 4.2, that we now re-formulate in our framework.

Depoissonization (reformulation from Theorem 4.2). *Let $G(Z)$ be the Poisson transform of a sequence $R(n)$, namely*

$$G(Z) = e^{-Z} \sum_{n \geq 0} R(n) \frac{Z^n}{n!}.$$

This series is assumed to be absolutely convergent for $z \in \mathbb{C}$. Assume moreover that the sequence $R(n)$ admits an analytic extension $Z \mapsto R(Z)$ in a cone S_α , which is of polynomial growth in this cone S_α when $|Z| \rightarrow \infty$, i.e., there exists γ such that, for $|Z| \rightarrow \infty$

$$|\arg(Z)| \leq \alpha \implies |R(Z)| = O(|Z|^\gamma).$$

Then, there exists a cone S_β where the following estimate holds for $|Z| \rightarrow \infty$,

$$G(Z) = R(Z) + \frac{Z}{2} R''(Z) + O(|Z|^{\gamma-2}).$$

An easy application of the previous result, with the estimates for $G(Z)$ given in (6.15) leads to the proof of Assertion (iv). This ends the proof of Proposition 6.11.

6.3.7 An alternative proof for QuickSort and InsSort.

It is clear that two algorithms of the studied family —QuickSort and InsSort— are strongly faithful whereas the last one BubSort is not strongly faithful.

In the case of the two faithful algorithms, the following result easily follows from Proposition 6.11 and Relation (6.6). We only provide here the asymptotic main terms.

Theorem 6.12. *Consider a strongly faithful algorithm which sorts words that are independently drawn from the same source, assumed to be Λ -tame.*

- (i) *If the mean number $K(n)$ of key comparisons is $An^2 + O(n)$, then the mean number of symbol comparisons satisfies $S(n) = Ac(\mathcal{S})n^2 + O(n \log n)$, and $S(n)/K(n)$ is asymptotic to $c(\mathcal{S})$.*
- (ii) *If the mean number $K(n)$ of key comparisons is $An \log n + O(n)$, then the mean number $S(n)$ of symbols comparisons is asymptotic to $A/(2h(\mathcal{S})) \cdot n \log^2 n$ and $S(n)/K(n)$ is asymptotic to $\log n/(2h(\mathcal{S}))$.*

Then, the approach of Seidel, combined with our methods, provides an alternative approach for our main theorem, at least for the algorithms **QuickSort** and **InsSort**. However, this approach cannot be applied to **BubSort** that is not strongly faithful.

6.4 An asymptotic lower bound for $S(n)$.

Combining our methods described in Proposition 6.11 together with the approach of Seidel for lower bounds, we obtain an asymptotic lower bound $\underline{S}(n)$ for the mean number of symbol comparisons for any sorting algorithm (not necessarily faithful) using the standard string comparison procedure and dealing with words of a Λ -tame source \mathcal{S} . This result was already stated as Theorem 6.5.

6.4.1 Proof of Theorem 6.5.

We use the same notations as in Section 6.3.1. We consider a set \mathcal{U} of n distinct words, and a key-comparison based sorting algorithm \mathcal{A} . The set \mathcal{U} is presented as input to algorithm \mathcal{A} in order given by some permutation σ , and we denote by $\mathcal{K}[\mathcal{U}, \mathcal{U}_{(w)}, \sigma]$ the number of comparisons performed by \mathcal{A} on the subset $\mathcal{U}_{(w)}$ when \mathcal{U} is input under permutation σ .

We denote by L the function $L(n) = \log_2(n!)$, which appears in the lower bound for the mean number of key comparisons. We fix a subset \mathcal{U} and an algorithm \mathcal{A} and we say that a permutation σ is k -good for $\mathcal{V} \subset \mathcal{U}$ if $\mathcal{K}[\mathcal{U}, \mathcal{V}, \sigma] \geq L(|\mathcal{V}|) - k$. If it is not k -good, it is said to be k -bad. We will use the following lemma due to Seidel [65].

Lemma 6.13. [Seidel] *For any subset \mathcal{U} of cardinality n , and any algorithm \mathcal{A} , there is a set $\mathfrak{S} \subset \mathfrak{S}_n$ for which the following holds:*

- (i) *The cardinality $|\mathfrak{S}|$ satisfies: $|\mathfrak{S}| \geq n! (1 - (1/n))$;*
- (ii) *All the elements of \mathfrak{S} are $(2 \log n)$ -good for any $\mathcal{U}_{(w)}$.*

We will prove the Lemma later on. We first explain how it entails the proof of Theorem 6.5. Indeed, with the second relation of Eq. (6.4), Lemma 6.13 entails the inequality, for any set \mathcal{U} of cardinality n ,

$$S[\mathcal{U}] \geq \left(1 - \frac{1}{n}\right) (R_1[\mathcal{U}] - 2 \log n R_2[\mathcal{U}]),$$

where the parameters R_1 and R_2 are respectively associated with the toll functions $f(k) = L(k) := \log_2(k!)$ and $f(k) = 1$. Now, Proposition 6.11 provides the asymptotic behaviour for the mean values $R_1(n)$ and $R_2(n)$, namely

$$R_1(n) \sim \frac{1}{2 \log 2} \frac{1}{h(\mathcal{S})} n \log^2 n, \quad R_2(n) = \Theta(n).$$

This proves that the mean number of symbol comparisons of the algorithm \mathcal{A} admits the asymptotic lower bound

$$S(n) \geq \underline{S}(n), \quad \text{with} \quad \underline{S}(n) \sim \frac{1}{2 \log 2} \frac{1}{h(\mathcal{S})} n \log^2 n,$$

and Theorem 6.5 is proven.

6.4.2 Proof of Lemma 6.13

We follow Seidel and use decision trees that have been described in Section 1.1.5, and we copy here as Figure 6.10 the figure which already appeared in Chapter 1.

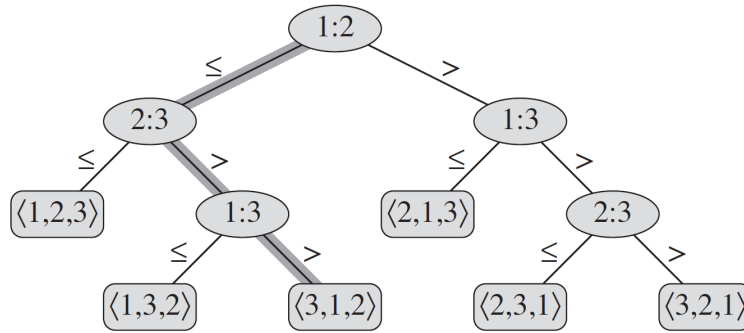


Figure 6.10: Decision tree of the Insertion Sort Algorithm when operating on three elements

Consider the decision tree \mathcal{D} associated with the algorithm \mathcal{A} . The set $\mathcal{U}_{\langle w \rangle}$ is an order contiguous subrange of \mathcal{U} , i.e., $\mathcal{U}_{\langle w \rangle} = \mathcal{U}_{[i..j]}$. Consider the set of permutations $\underline{\mathfrak{S}}$ whose restriction to the set $[1..n] \setminus [i..j]$ is fixed. Thus $|\underline{\mathfrak{S}}| = N_w!$. Each leaf of \mathcal{D} corresponds to a permutation σ . Take the leaves that correspond to permutations in $\underline{\mathfrak{S}}$ along with their rootpaths. They induce a subtree of \mathcal{D} . We contract all the paths in this tree by removing all the non branching nodes; there results a binary tree $\underline{\mathcal{D}}$ that represents a valid decision tree for a sorting algorithm on $\mathcal{U}_{\langle w \rangle}$. Since $\underline{\mathcal{D}}$ is a binary tree with $N_w!$ leaves, for any $k > 0$, there can be at most a $1/2^k$ -fraction of these leaves that have distance less than $L(N_w) - k$ from the root. In other words, at most a $1/2^k$ -fraction of the permutations $\sigma \in \underline{\mathfrak{S}}$ are k -bad for $\mathcal{U}_{\langle w \rangle}$. Since this is true for any way the permutation values outside $[i, j]$ were fixed, we get that for any $k > 0$, the fraction of all permutations that are k -bad for $\mathcal{U}_{\langle w \rangle}$ is at most 2^{-k} .

Now, we observe that, although the trie $T(\mathcal{U})$ can have arbitrarily many nodes w , there are only at most $n - 1$ different sets $\mathcal{U}_{\langle w \rangle}$. There is a clear equality between the number of different sets $\mathcal{U}_{\langle w \rangle}$ and the number of branching nodes in the trie. We now prove that the number of branching nodes in a trie with n leaves is at most $n - 1$, with an easy recursion. For $n = |\mathcal{U}| = 2$, there is at most 1 branching node. For a cardinality $n := |\mathcal{U}| \geq 2$, we consider the first branching node in the trie (the one with the smallest level). Then, each subtree of cardinality $n_i < n$ has at most $n_i - 1$ branching nodes, and there are at most $1 + \sum_{i \in \Sigma} (n_i - 1) \leq n - 1$ branching nodes.

Thus choosing $k \geq 2 \log n$ ensures that there is a subset \mathfrak{S} of \mathfrak{S}_n whose cardinality is at least $(1 - n2^{-k})n! \geq [1 - (1/n)]n!$ and whose elements are k -good for all the subsets $\mathcal{U}_{\langle w \rangle}$. This ends the proof of Lemma 6.13.

6.4.3 Relation between various lower bounds.

The well-known lower bound $\underline{K}(n)$ for any sorting algorithm using key comparisons is asymptotic to $n \log_2 n$. Then, we have proven that

$$\frac{\underline{S}(n)}{\underline{K}(n)} \sim \frac{1}{2h(\mathcal{S})} \log n, \quad (6.16)$$

and this is the same ratio as the ratio which appears in (6.2) for the non-robust algorithms, namely **QuickSort** and **BubSort**.

There is also a classical lower bound in information theory which states that the number $D(n)$ of symbol comparisons used by any sorting algorithm on words which uses the symbol representation of words satisfies

$$D(n) \geq \underline{D}(n), \quad \text{with} \quad \underline{D}(n) \sim \frac{1}{h(\mathcal{S})} n \log n,$$

and the following ratio between the two asymptotic lower bounds for sorting words holds

$$\frac{\underline{S}(n)}{\underline{D}(n)} \sim \frac{1}{2 \log 2} \log n.$$

Conclusion of the Chapter. We have described two main results of the thesis about sorting algorithms, namely Theorem 6.2 which provides our results on the three algorithms **QuickSort**, **InsSort** and **BubSort**, together with Theorem 6.5 which provides a lower bound. We also gave an alternative approach to the proof of **QuickSort** and **InsSort** using Seidel's point of view. In the following Chapter, we apply our general framework to **QuickSelect**, a typical searching algorithm using “divide and conquer” principle, and its variants like **QuickMin** and **QuickMax**. We will see that it is not possible to use our general method for **Quickselect**. We will use an indirect way, via a new algorithm **QuickVal**. We will use the Laplace method to prove that **Quickselect** and **QuickVal** have the same asymptotic behavior.

Chapter 7

Towards a realistic analysis of the QuickSelect algorithm.

Contents

7.1	Results	124
7.1.1	Tameness properties needed for the source.	124
7.1.2	General result.	125
7.1.3	Constants of the analysis.	125
7.1.4	Particular case of the binary source.	126
7.2	Algebraic analysis.	128
7.2.1	Generalities	128
7.2.2	When the keys are words.	128
7.2.3	The QuickVal algorithm	129
7.2.4	Poissonized density.	130
7.3	Expectation in the Poisson model.	131
7.3.1	Exact formulae for the mean number of symbol comparisons in the Bernoulli model.	131
7.3.2	The case of the algorithm QuickQuant.	132
7.3.3	General principles for the Asymptotic Analysis	133
7.4	Asymptotic Analysis of QuickVal	133
7.4.1	Analytic lifting and Rice formula.	133
7.4.2	Asymptotic estimates for the mean number of symbol comparisons for QuickMin, QuickMax, QuickVal and QuickRand.	134
7.4.3	Tameness of the mixed series $\varpi(s)$	134
7.4.4	Principles of the proof of Proposition 7.12.	135
7.4.5	Study of integrals over triangles.	136
7.4.6	Study of the integral over the rectangle \mathcal{R}	137
7.4.7	End of the proof.	139
7.5	Return to the analysis of the QuickQuant_{α} Algorithm.	139
7.5.1	Regularity of the asymptotic slope $\alpha \mapsto \rho(\alpha)$ of QuickVal.	139
7.5.2	Return to the analysis of QuickQuant.	141
7.5.3	Laplace's method.	142
7.5.4	A tailored version of the Laplace method.	143
7.5.5	Proof of Theorem 7.16.	144

7.6	Study of the asymptotic slope $\rho(\alpha)$	146
7.6.1	An alternative expression for the asymptotic slope $\alpha \mapsto \rho(\alpha)$ of QuickVal.	146
7.6.2	Maximality of the asymptotic slope of the QuickVal algorithm at $\alpha = 1/2$.	147

This chapter is devoted to the analysis of the **QuickSelect** algorithm and its variants. The **QuickSelect**(m, n) algorithm is based on the “Divide and Conquer” principle and returns the key of rank m of an input array of n keys. As **QuickSort**, it uses the first key of the array as a pivot and performs the partition operation (see the description of the algorithm in Section 1.3). If the rank k of the pivot equals m , the algorithm returns the pivot. If $k > m$, the algorithm continues with the left sub-array; otherwise, it continues with the right sub-array. We shall mostly focus our attention on situations where the rank m is proportional to n , being of the form $m = \lfloor 1 + \alpha(n - 1) \rfloor$, for some $\alpha \in [0, 1]$, so that the algorithm determines the α -th quantile; it will then be denoted by **QuickQuant** $_{\alpha}(n)$. For $\alpha = 0$, the rank m equals 1 and this is **QuickMin**(n). For $\alpha = 1$, the rank m equals n and this is **QuickMax**(n). We also consider the case where m is randomly chosen in $[1..n]$, and this is **QuickRand**(n): this last algorithm is of course artificial and is defined to model an “average” **QuickSelect** procedure.

It is possible to directly apply the general method of Chapter 5 to **QuickMin** or **QuickMax**. But this is not the case of **QuickSelect** for a general rank. This is why we introduce a new algorithm, the **QuickVal** algorithm, which can be seen as a dual version of **QuickSelect**. This algorithm is analyzed with the general method of Chapter 5, and we explain how we can return from the analysis of **QuickVal** to the analysis of the **QuickSelect** Algorithm, its dual counterpart.

7.1 Results

Our main result stated in Theorem 7.2, describes the case of the **QuickSelect** algorithm with all its variants [15]. It involves various constants that depend on the source \mathcal{S} (and possibly on the parameter $\alpha \in [0, 1]$). These are precisely described in Theorem 7.3 (with some plots displayed in Figure 7.1) and obtained in Section 7.4.3. Our results hold under tameness conditions that we now describe.

7.1.1 Tameness properties needed for the source.

As we already announced in Section 3 of Chapter 5, we need tameness properties for the source \mathcal{S} , which are not of the same type as for Sorting algorithms. In fact, we need here weakest notions of tameness which involve the two series Π and Λ , whose definition is recalled here:

$$\Lambda(s) := \sum_{w \in \Sigma^*} p_w^s = \sum_{k \geq 0} \Lambda_k(s), \quad \text{with } \Lambda_k(s) := \sum_{w \in \Sigma^k} p_w^s, \quad (7.1)$$

$$\Pi(s) := \sum_{k \geq 0} \pi_k^s, \quad \text{where } \pi_k := \max\{p_w : w \in \Sigma^k\}. \quad (7.2)$$

We will see in Section 7.4.3 that the mixed Dirichlet series $\varpi(s)$ relative to **QuickSelect** studies may be related to these two Dirichlet series Λ and Π .

Two types of properties of the source may entail tameness for the mixed series $\varpi(s)$. They involve the weak tameness of the Dirichlet series Λ or Π , as it is defined in Section 3.3

(Chapter 3). The Dirichlet series Π does not intervene in the analyses of sorting algorithms in Section in Chapter 6: only the Λ series plays a central role. We recall there this definition, already given in Chapter 3.

Definition 7.1. *A source is weakly Λ -tame with width δ if the abscissa of absolute convergence of the Dirichlet series $\Lambda(s)$ is equal to $1 + \delta$.*

A source is weakly Π -tame with width δ if the abscissa of absolute convergence of the Dirichlet series $\Pi(s)$ is equal to δ .

Furthermore, as the two series Λ and Π satisfy the relations

$$|\Pi(s)| \leq \Lambda(\Re s) \quad (\text{for } \Re s \geq 0) \quad \text{and} \quad |\Lambda(s)| \leq \Pi(\Re s - 1) \quad (\text{for } \Re s \geq 1), \quad (7.3)$$

this entails the inequality $\delta(\Pi) \geq \delta(\Lambda)$ between the two widths $\delta(\Pi)$ and $\delta(\Lambda)$.

7.1.2 General result.

Theorem 7.2. *For a source \mathcal{S} both Π -tame and weakly Λ -tame with a small enough width, the mean number of symbol comparisons of all the variants of **QuickSelect**(n) is $\Theta(n)$. More precisely, the following holds:*

- (a) *If the source is Π -tame with width $\delta_0 < 1/2$, then the mean number of symbol comparisons $QQ_\alpha(n)$ for **QuickQuant** $_\alpha$ satisfies, for $\delta < 1 - 2\delta_0$, and for any $\alpha \in [0, 1]$,*

$$QQ_\alpha(n) = \rho_S(\alpha)n + O(n^{1-\delta}),$$

where the constant hidden in the O -term is uniform (in α) for any interval of the form $[\alpha_1, 1 - \alpha_1]$ with $\alpha_1 > 0$.

- (b) *If the source is Π -tame with width $\delta_0 < 1/2$, the mean number $QR(n)$ of symbol comparisons performed by **QuickRand**(n) satisfies, for $\delta < 1 - 2\delta_0$,*

$$QR(n) = \gamma_S n + O(n^{1-\delta}), \quad \text{with} \quad \gamma_S = \int_0^1 \rho(\alpha) d\alpha.$$

- (c) *If the source is weakly Λ -tame with width $\delta_1 < 1/2$, the mean number of symbol comparisons, $QM_{(-)}(n)$ for **QuickMin**(n) and $QM_{(+)}(n)$ for **QuickMax**(n), satisfies, $\delta < 1 - 2\delta_1$, with $\epsilon = \pm$,*

$$QM_{(\epsilon)}(n) = \rho_S^{(\epsilon)} n + O(n^{1-\delta}), \quad \text{with} \quad \rho_S^{(+)} = \rho_S(1), \quad \rho_S^{(-)} = \rho_S(0).$$

7.1.3 Constants of the analysis.

The following result provides precise expressions for the constants which appear in the dominant terms (and corrects small errors in Figure 1 of [12]) in terms of the probabilities $p_w^{(-)}, p_w^{(+)}$ defined as

$$p_w^{(-)} := \sum_{\substack{\alpha: |\alpha|=|w|, \\ \alpha \prec w}} p_\alpha, \quad p_w^{(+)} := \sum_{\substack{\alpha: |\alpha|=|w|, \\ \alpha \succ w}} p_\alpha. \quad (7.4)$$

They are related to the ends of the fundamental intervals a_w, b_w and probabilities p_w usually used in the thesis and defined in Chapter 2 as

$$p_w^{(-)} = a_w, \quad p_w^{(+)} = 1 - b_w, \quad p_w = 1 - (p_w^{(+)} + p_w^{(-)}) . \quad (7.5)$$

Theorem 7.3. *For a general source \mathcal{S} , where the probabilities $p_w^{(+)}$, $p_w^{(-)}$, and $p_w = 1 - p_w^{(+)} - p_w^{(-)}$ are defined in Eq. (7.4), the constants of interest involved in Theorem 7.2 admit the following expressions:*

(a) *The quantile constant $\rho_{\mathcal{S}}(\alpha)$ is defined as*

$$\rho_{\mathcal{S}}(\alpha) := \sum_{w \in \Sigma^*} p_w \kappa \left(\frac{\alpha - p_w^{(-)}}{p_w} \right),$$

and involves the function κ given by $\kappa(y) := 2[1 + H(y)]$

$$H(y) := \begin{cases} y \log \left(1 - \frac{1}{y} \right) & \text{if } y < 0, \\ h(y) = -y \log y - (1 - y) \log(1 - y) & \text{if } y \in [0, 1], \\ (y - 1) \log \left(1 - \frac{1}{y} \right) & \text{if } y > 1. \end{cases}$$

Note that $H(y)$ is a continuous extension, symmetric with respect to $y = 1/2$, of the entropic function, using the entropy function $h(y) = -y \log y - (1 - y) \log(1 - y)$ to the whole real line.

(b) *The Random selection constant is*

$$\gamma_{\mathcal{S}} = \sum_{w \in \Sigma^*} p_w^2 \left[2 + \frac{1}{p_w} + \sum_{\epsilon = \pm} \left[\log \left(1 + \frac{p_w^{(\epsilon)}}{p_w} \right) - \left(\frac{p_w^{(\epsilon)}}{p_w} \right)^2 \log \left(1 + \frac{p_w}{p_w^{(\epsilon)}} \right) \right] \right].$$

(c) *The QuickMin/QuickMax constants are*

$$\rho_{\mathcal{S}}^{(\epsilon)} = 2 \sum_{w \in \Sigma^*} p_w \left[1 - \frac{p_w^{(\epsilon)}}{p_w} \log \left(1 + \frac{p_w}{p_w^{(\epsilon)}} \right) \right].$$

Note that the κ function—at least its restriction to the $[0, 1]$ interval that coincides with the entropy function—already intervenes in the mean number $K(n)$ of key comparisons of the **QuickQuant** algorithm, as recalled in Section 1.3.2. Figure 7.1 shows the graphs of the function $\alpha \mapsto \rho(\alpha)$ for different Bernoulli sources, together with the graph of the κ function.

It appears that the graph of ρ is a deformation of the graph of κ , and the plots illustrate the fractal character of the constants involved in **QuickQuant**, as the expression for ρ in terms of κ suggests. We see that the function ρ is not always maximized at $\alpha = 1/2$, not even for symmetric sources, even though κ is maximized at $1/2$. There is then a natural question to ask: For which (symmetric) sources is the maximum for ρ attained at $\alpha = 1/2$? If the maximum is not attained at $\alpha = 1/2$, at which point is it attained?

7.1.4 Particular case of the binary source.

All these constants specialize nicely for the *standard binary source* \mathcal{B} (when keys are compared via their binary expansions, with uniform independent bits), in which case they admit pleasant expressions that simplify and extend those of Fill and Nakama [23] and Grabner and Prodinger [35] and lead themselves to precise numerical evaluations.

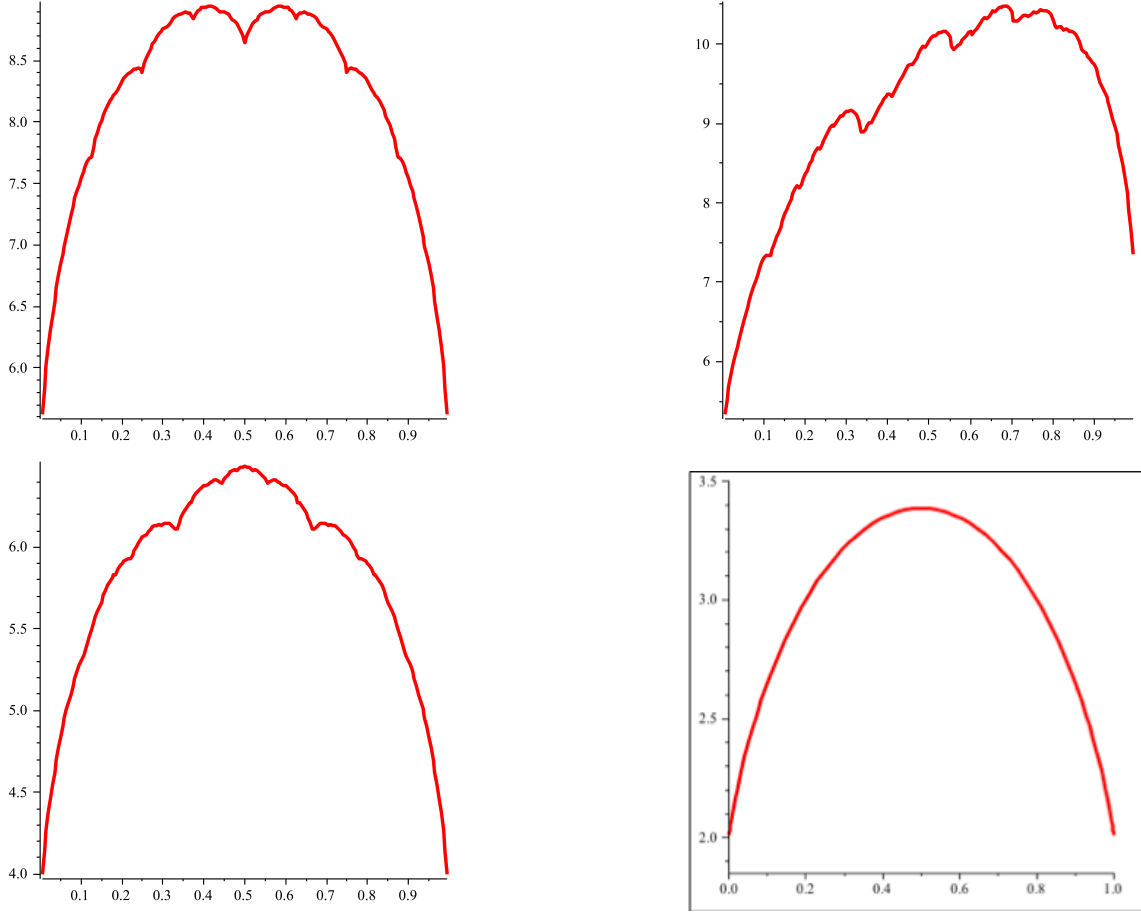


Figure 7.1: Plots of the function $\rho_S(\alpha)$ for $\alpha \in [0, 1]$ and three sources: $\mathcal{B}(1/2, 1/2)$, $\mathcal{B}(1/3, 2/3)$, and $\mathcal{B}(1/3, 1/3, 1/3)$. The curves are deformations of the curve $\alpha \mapsto \kappa(\alpha)$ whose plot is on the bottom-right corner. The plots illustrate the fractal character of the constants involved in **QuickQuant**.

Proposition 7.4. *For a binary source, the constants admit the following expressions:*

$$\rho_{\mathcal{B}}(\alpha) = \sum_{\ell \geq 0} \frac{1}{2^\ell} \sum_{k=0}^{2^\ell-1} \kappa(2^\ell \alpha - k)$$

$$\rho_{\mathcal{B}}^{(\epsilon)} = 4 + 2 \sum_{\ell \geq 0} \frac{1}{2^\ell} \sum_{k=1}^{2^\ell-1} \left[1 - k \log \left(1 + \frac{1}{k} \right) \right] \doteq 5.27937\,82410\,80958$$

$$\gamma_{\mathcal{B}} = \frac{14}{3} + 2 \sum_{\ell=0}^{\infty} \frac{1}{2^{2\ell}} \sum_{k=1}^{2^\ell-1} \left[k + 1 + \log(k+1) - k^2 \log \left(1 + \frac{1}{k} \right) \right] \doteq 8.20730\,88638.$$

Here, the function κ is defined in Theorem 7.3(a).

7.2 Algebraic analysis.

7.2.1 Generalities

We recall our results of Chapter 1: in the **QuickSelect** algorithm, as in the **QuickSort** algorithm, the keys U_i and U_j are compared if and only if U_i or U_j is the first key among the subset $\mathcal{U}_{[k,\ell]}$, with $k := i \wedge m$, $\ell := j \vee m$. Then, the mean number of key comparisons between U_i and U_j performed by **QuickSelect**(m) is, for $i < j$, equal to

$$\pi_n^{(m)}(i, j) = \frac{2}{\text{card } \mathcal{U}_{[k,\ell]}} = \frac{2}{\ell - k + 1}, \quad k := i \wedge m, \quad \ell := j \vee m.$$

Observe that this probability depends on both i and j , except when $m = 1$ (case of **QuickMin**) or $m = n$ (case of **QuickMax**), where it satisfies

$$\pi_n^{(1)}(i, j) = \frac{2}{\text{card } \mathcal{U}_{[1,j]}} = \frac{2}{j}, \quad \pi_n^{(n)}(i, j) = \frac{2}{\text{card } \mathcal{U}_{[i,n]}} = \frac{2}{n - i + 1}.$$

We are also interested in two variants of the **QuickSelect**(m, n) algorithm:

- (a) The **QuickQuant** $_{\alpha}(n)$ algorithm outputs the α -quantile (with $\alpha \in [0, 1]$), namely, the key of rank $\lfloor 1 + \alpha(n - 1) \rfloor$. With this definition, we remark that **QuickQuant** $_0(n)$ coincides with **QuickMin**(n), whereas **QuickQuant** $_1(n)$ coincides with **QuickMax**(n).
- (b) The **QuickRand**(n) algorithm outputs a key whose rank is chosen uniformly at random in the interval $[1 \dots n]$.

7.2.2 When the keys are words.

Here, the keys are words, and we adopt the following general model for the set of inputs: we consider a finite sequence $\mathcal{V} = (V_1, \dots, V_n)$ of infinite words independently produced by the same source \mathcal{S} . Such a sequence \mathcal{V} is obtained with a sequence $\mathbf{x} = (x_1, x_2, \dots, x_n)$ formed by n independent drawings x_1, x_2, \dots, x_n in the interval $\mathcal{I} = [0, 1]$ via the mapping M , and we set $V_i := M(x_i)$. Inside this model, the words $M(x_i)$ are distinct with probability 1.

We assume moreover that \mathcal{V} contains two given words $M(u)$ and $M(t)$, with $u < t$. We first consider the number of key comparisons between $M(u)$ and $M(t)$, and deal with the mean number $\hat{\pi}[u, t]$ of key comparisons performed by the algorithm between two words $M(u)$ and $M(t)$ *chosen as keys*, where the mean is taken with respect to all the permutations of \mathcal{V} .

For $(x, y) \in [0, 1]^2$ with $x < y$, the variables $N_{[x,y]}$, $N_{]x,y[}$ respectively denote the number of words of \mathcal{V} whose parameter belong to $[x, y]$ or $]x, y[$. In Section 1.3.2, the mean number $\pi(i, j)$ depends on the cardinality of sets of the form $\mathcal{U}_{[\ell,k]}$ for various pairs (ℓ, k) . The present study is based on the variables $N_{[x,y]}$ or $N_{]x,y[}$, and, in the same vein as in Section 1.3.2, the mean number $\hat{\pi}(u, t)$ depends on the cardinalities $N_{[x,y]}$ or $N_{]x,y[}$ for various pairs $(x, y) \in [0, 1]^2$ related to (u, t) and possibly α in the case of **QuickQuant** $_{\alpha}$.

QuickMin and QuickMax. When $M(u)$ and $M(t)$ are chosen as keys, the variable $N_{[0,t]}$ satisfies the equality $N_{[0,t]} = 2 + N_{[0,u]} + N_{]u,t[}$. In the same vein, the variable $N_{[u,1]}$ satisfies the equality $N_{[u,1]} = 2 + N_{]u,t[} + N_{[t,1]}$, and then

$$\hat{\pi}_-(u, t) = \frac{2}{N_{[0,t]}} = \frac{2}{N_{[0,u]} + N_{]u,t[} + 2}, \quad \hat{\pi}_+(u, t) = \frac{2}{N_{[u,1]}} = \frac{2}{N_{]u,t[} + N_{[t,1]} + 2}. \quad (7.6)$$

QuickSelect. For the **QuickSelect**(m) algorithm, we assume that \mathcal{V} contains two given words $M(u)$ and $M(t)$, with $u < t$. We denote by α the parameter of the word of rank m . The mean number of comparisons $\hat{\pi}_\alpha(u, t)$ performed by the **QuickSelect** algorithm between the words $M(u)$ and $M(t)$ *chosen as keys* when it searches the word $M(\alpha)$ is

$$\hat{\pi}_\alpha(u, t) = \frac{2}{N_{[x, y]}} \quad \text{with } x := u \wedge \alpha, y := t \vee \alpha.$$

We denote by z the “middle point” (i.e., $x \leq z \leq y$), so that the equality $\{x, y, z\} = \{\alpha, u, t\}$ holds.

- (i) In the case when $\alpha \neq u$ and $\alpha \neq t$, the three words $M(x)$, $M(y)$ and $M(z)$ are distinct and, in the same vein as previously, $N_{[x, y]} = N_{[x, z]} + N_{[z, y]} + 3$ where the translations of 3 expresses that the three words $M(x)$, $M(y)$, $M(z)$ belong to \mathcal{V} .
- (ii) In the case when $\alpha = u$ or $\alpha = t$, there are only two distinct words amongst $M(x)$, $M(y)$ and $M(z)$ and, $N_{[x, y]} = N_{[x, z]} + N_{[z, y]} + 2$ where the translation of 2 expresses that there are only two distinct words.

Finally, with $x := u \wedge \alpha$, $y := t \vee \alpha$, $\{x, y, z\} = \{\alpha, u, t\}$, we write

$$\hat{\pi}_\alpha(u, t) = \begin{cases} \frac{2}{N_{[x, z]} + N_{[z, y]} + 3}, & (u \neq \alpha \text{ and } t \neq \alpha); \\ \frac{2}{N_{[x, z]} + N_{[z, y]} + 2}, & (u = \alpha, \text{ or } t = \alpha). \end{cases} \quad (7.7)$$

7.2.3 The QuickVal algorithm

We remark that Eq. (7.7) does not provide a clear tool for computing the mean number of comparisons $\hat{\pi}(u, t)$ of the **QuickSelect**(m) algorithm for a general rank m . This is due to the fact that the interval $[x, y]$ depends on the parameter α of the word of rank m , which is not directly related to the rank m . This is why we consider a dual algorithm of **QuickSelect** that admits α as the input of the algorithm. That is called **QuickVal**.

QuickVal takes as input a set of words \mathcal{V} and a given *word* $V \in \mathcal{V}$, and returns the *rank* of V inside the set \mathcal{V} . This algorithm is of independent interest and is easily implemented as a variant of **QuickSelect** by resorting to the usual partitioning phase, then doing a comparison between the *value* of the pivot and the input *word* V (rather than a comparison between their ranks). We call this algorithm **QuickVal** $_\alpha$ when it is used to seek the rank of the word $M(\alpha)$. By definition, the two algorithms **QuickSelect**(m) and **QuickVal** $_\alpha$ behave exactly in the same way when the rank of the word $M(\alpha)$ equals m . Now, with (7.7), one computes the mean number of key comparisons $\hat{\pi}_\alpha(u, t)$ performed by **QuickVal** $_\alpha$ between two words $M(u)$ and $M(t)$.

If we consider the algorithm **QuickQuant** $_\alpha(n)$ which outputs the key of rank $\lfloor 1 + (n - 1)\alpha \rfloor = \lfloor n\alpha + 1 - \alpha \rfloor$, then, for $\alpha = 0$, we recover **QuickMin**, and, for $\alpha = 1$, we recover **QuickMax**. For a general $\alpha \in]0, 1[$, the behaviors of the two algorithms of **QuickVal** $_\alpha(n)$ and **QuickQuant** $_\alpha(n)$ should be asymptotically similar. Indeed, the parameter of the α -*quantile* of a random set of words of large enough cardinality must be, with high probability, close to α . It is proven in [52] that this parameter follows asymptotically a Gaussian law with mean α and variance $\alpha(1 - \alpha)/n$. We will return to this question in Section 7.3.2.

```

Function QuickSel( $V, 1, n, m$ )
  /* Returns the element of rank  $m$  of the array  $V[1..n]$  */
   $k \leftarrow \text{Partition}(V, 1, n)$ 
  if  $k > m$  then
    | return QuickSel( $V, 1, k - 1, m$ )
  else
    | if  $k < m$  then
      | return QuickSel( $V, k, n, m - k$ )
    | else
      | return  $V[k]$ 
    | end
  end

```

```

Function QuickVal( $V, 1, n, v$ )
  /* Returns the rank of the element  $v \in V$  */
   $k \leftarrow \text{Partition}(V, 1, n)$ 
  if  $V[k] > v$  then
    | return QuickVal( $V, 1, k - 1, v$ )
  else
    | if  $V[k] < v$  then
      | return QuickVal( $V, k, n, v$ )
    | else
      | return  $k$ 
    | end
  end

```

Figure 7.2: The two algorithms QuickSel and QuickVal. The Partition procedure is discussed in Section 1.2.

7.2.4 Poissonized density.

We are first interested in the density $\Phi_Z(u, t)$ of the three algorithms, that is computed via its coefficients $\varphi(k, u, t)$.

Lemma 7.5. *For each algorithm QuickMin, QuickMax and QuickVal $_\alpha$, the sequence $\varphi(k, u, t)$ satisfies for $k \geq 2$,*

$$\begin{aligned}
 \varphi_-(k, u, t) &= 2t^{k-2} \\
 \varphi_+(k, u, t) &= 2(1-u)^{k-2} \\
 \varphi_\alpha(k, u, t) &= \begin{cases} \frac{4}{k+1}(t \vee \alpha - u \wedge \alpha)^{k-2} & (u \neq \alpha \text{ and } t \neq \alpha) \\ 2(t \vee \alpha - u \wedge \alpha)^{k-2} & (u = \alpha, \text{ or } t = \alpha). \end{cases}
 \end{aligned}$$

Proof. This is an (easy) application of Proposition 5.5. First, we observe the decompositions in the “basis” $G_m(X)$ of Eq. (5.8)

$$\frac{1}{X+2} = G_1(X) - g_2(X), \quad \frac{1}{X+3} = 2G_3(X) - 2G_2(X) + G_1(X).$$

Then, we rewrite the density according to 5.3 for QuickMin (using $\lambda = t$) and the density for QuickMax (using $\lambda = 1 - u$)

$$\Phi_Z(u, t) = 2Z^2 [G_1(Zt) - G_2(Zt)], \quad \Phi_Z(u, t) = 2Z^2 [G_1(Z(1-u)) - G_2(Z(1-u))].$$

For QuickVal_α , we use $\lambda = y - x$ with $x = u \wedge \alpha, y = t \vee \alpha$, and we obtain

$$\Phi_Z(u, t) = \begin{cases} 2Z^2 [2G_3(Z(y - x)) - 2G_2(Z(y - x)) + G_1(Z(y - x))], & (t \neq \alpha \text{ and } u \neq \alpha) \\ 2Z^2 [G_1(Z(y - x)) - G_2(Z(y - x))], & (t = \alpha, \text{ or } u = \alpha) \end{cases}$$

Then, applying Proposition 5.5 ends the proof of the Lemma. \square

7.3 Expectation in the Poisson model.

The coefficient $\varphi(k)$ is easy to compute with integrals of $\varphi(k, u, t)$ on the triangles \mathcal{T}_w . It depends both on the algorithm (via the sequence of functions $\varphi(k, u, t)$) and the source (via the fundamental triangles \mathcal{T}_w). In Lemma 7.5, for the QuickVal algorithm, there are two expressions for $\varphi_\alpha(k, u, t)$, and the first one holds almost everywhere on the triangle \mathcal{T} , except on the reunion of the two lines $u = \alpha$ and $t = \alpha$ whose contribution is zero in the integral.

Lemma 7.6. *For each algorithm QuickMin , QuickMax and QuickVal_α , the sequence $\varphi(k)$ satisfies for $k \geq 2$ respectively*

$$\varphi_-(k) = 2\mathcal{J}[t^{k-2}], \quad \varphi_+(k) = 2\mathcal{J}[(1 - u)^{k-2}], \quad \varphi_\alpha(k) = \frac{4}{k+1} \mathcal{J}[(t \vee \alpha - u \wedge \alpha)^{k-2}].$$

7.3.1 Exact formulae for the mean number of symbol comparisons in the Bernoulli model.

It is easy to return to the Bernoulli model $(\mathcal{B}_n, \mathcal{S})$, where we are interested in the mean number $S(n)$ of symbol comparisons performed by the algorithm. The mean number $S(n)$ of symbol comparisons used by the algorithm when it deals with n words independently drawn from the same source is related to S_Z and then to $\varphi(n)$ by the equalities

$$S(n) = \sum_{k=2}^n (-1)^k \binom{n}{k} \varphi(k), \quad (7.8)$$

which provide an exact formula for the mean number $S(n)$ of symbol comparisons, in the case of QuickVal_α (denoted by $\text{QV}_\alpha(n)$).

Proposition 7.7. *The mean number of symbol comparisons for each algorithm of interest, $\text{QM}_\pm(n)$ for QuickMin and QuickMax , $\text{QV}_\alpha(n)$ for $\text{QuickVal}_\alpha(n)$ admits an exact expression which involves the functional \mathcal{J} of the source (see Eq. (2.17)) recalled here*

$$\mathcal{J}[\Phi] = \sum_{w \in \Sigma^\star} \int_{\mathcal{T}_w} g(u, t) du dt.$$

One has respectively

$$\text{QM}_-(n) = 2 \sum_{k=2}^n (-1)^k \binom{n}{k} \mathcal{J}[t^{k-2}], \quad (7.9)$$

$$\text{QM}_+(n) = 2 \sum_{k=2}^n (-1)^k \binom{n}{k} \mathcal{J}[(1 - u)^{k-2}], \quad (7.10)$$

$$\text{QV}_\alpha(n) = 4 \sum_{k=2}^n \frac{(-1)^k}{k+1} \binom{n}{k} \mathcal{J}[(t \vee \alpha - u \wedge \alpha)^{k-2}]. \quad (7.11)$$

7.3.2 The case of the algorithm QuickQuant.

We now relate the mean number of symbol comparisons $QQ_\alpha(n)$ of the algorithm $\text{QuickQuant}_\alpha(n)$ to the mean number of symbol comparisons $QV_\alpha(n)$ of the algorithm $\text{QuickVal}_\alpha(n)$. For a sequence $\mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathcal{I}^n$, the algorithm $\text{QuickQuant}(\alpha, n, \mathbf{x})$ outputs the word whose parameter is the real of rank $\lfloor 1 + \alpha(n - 1) \rfloor = \lfloor \alpha n + 1 - \alpha \rfloor$ in the sequence \mathbf{x} , denoted by $R(\alpha, n, \mathbf{x})$. The two algorithms $\text{QuickQuant}(\alpha, n, \mathbf{x})$ and $\text{QuickVal}(R(\alpha, n, \mathbf{x}), n, \mathbf{x})$ behave exactly in the same way. This is why we have to study, for each α and n fixed, the distribution of the random variable $[0, 1]^n \rightarrow [0, 1]$, defined as $\mathbf{x} \mapsto R(\alpha, n, \mathbf{x})$. This variable is just the statistics of order $m := \lfloor \alpha n + 1 - \alpha \rfloor$, and its density $f_n(\alpha, v)$ is the Beta law of parameters $(m, n - m + 1)$ defined as

$$f_n(\alpha, v) = \frac{n!}{(m-1)!(n-m)!} v^{m-1} (1-v)^{n-m}, \quad \text{with } m := \lfloor \alpha n + 1 - \alpha \rfloor. \quad (7.12)$$

This means that $f_n(\alpha, v)dv$ is the probability that $R(\alpha, n, \mathbf{x})$ belongs to the interval $[v, v + dv]$.

On the other hand, if we denote by $QQ(\alpha, n, \mathbf{x})$ and $QV(\alpha, n, \mathbf{x})$ the mean number of symbol comparisons of each algorithm on the sequence \mathbf{x} (the mean is taken over all the permutations of \mathbf{x}), the two mean numbers $QQ_\alpha(n)$, $QV_\alpha(n)$ are defined as

$$QQ_\alpha(n) = \int_{\mathcal{I}^n} QQ(\alpha, n, \mathbf{x}) d\mathbf{x}, \quad QV_\alpha(n) = \int_{\mathcal{I}^n} QV(\alpha, n, \mathbf{x}) d\mathbf{x}.$$

Then, there is an exact integral formula for $QQ_\alpha(n)$, namely,

$$\begin{aligned} QQ_\alpha(n) &= \int_{\mathcal{I}^n} QQ(\alpha, n, \mathbf{x}) d\mathbf{x} \\ &= \int_{\mathcal{I}^n \times \mathcal{I}} QV(v, n, \mathbf{x}) f_n(\alpha, v) dv d\mathbf{x} \\ &= \int_{\mathcal{I}} QV_v(n) f_n(\alpha, v) dv. \end{aligned} \quad (7.13)$$

The mean number $QS(m, n)$ of symbol comparisons of the algorithm $\text{QuickSelect}(m, n)$ is then related to the Beta law $\beta(m, n - m + 1)$ via the equality

$$\begin{aligned} QS(m, n) &= \int_{\mathcal{I}} QV_v(n) \beta(m, n - m + 1)(v) dv \\ &= \frac{n!}{(m-1)!(n-m)!} \int_{\mathcal{I}} QV_v(n) v^{m-1} (1-v)^{n-m} dv. \end{aligned} \quad (7.14)$$

The mean number of symbol comparisons $QR(n)$ performed by the QuickRand Algorithm on n words is equal to

$$QR(n) = \frac{1}{n} \sum_{m=1}^n QS(m, n),$$

and involves the mean of the densities $\beta(m, n - m + 1)$ which can be written as a binomial sum,

$$\frac{1}{n} \sum_{m=1}^n \beta(m, n - m + 1)(v) = \sum_{i=1}^n \frac{(n-1)!}{(m-1)!(n-m)!} v^{m-1} (1-v)^{n-m} = 1.$$

This proves the equality

$$QR(n) = \int_{\mathcal{I}} QV_v(n) dv.$$

Proposition 7.8. *The mean number of symbol comparisons $QQ_\alpha(n)$ for $\text{QuickQuant}_\alpha(n)$, and $QR(n)$ for QuickRand admits an exact expression which involves the functional \mathcal{J} of the source and the density of the Beta law of (7.12)*

$$QQ_\alpha(n) = \int_{\mathcal{I}} f_n(\alpha, v) QV_\alpha(n) d\alpha \quad (7.15)$$

$$QR(n) = 4 \sum_{k=2}^n \frac{(-1)^k}{k+1} \binom{n}{k} \int_{\mathcal{I}} \mathcal{J}[(t \vee \alpha - u \wedge \alpha)^{k-2}] d\alpha. \quad (7.16)$$

Note that our principle applies and, when we replace the functional \mathcal{J} by the integral \mathcal{L} on the triangle \mathcal{T} , we obtain a formula for the number of key comparisons for each algorithm.

7.3.3 General principles for the Asymptotic Analysis

We now explain how to obtain the asymptotic analysis for the variants of the **QuickSelect** algorithm. There are two cases:

- (a) For the algorithms **QuickVal** (and its variants), the methods of Chapter 5 can directly be applied, and this is done in Section 7.4. The mixed series $\varpi(s)$ admits an exact expression, and can be directly related to the series $\Lambda(s)$ and $\Pi(s)$. We exhibit such relations in Section 7.4.3, as we see in the statements of Lemmas 7.13 and 7.14. Then, we use the weak tameness of series Λ, Π , as defined in Section 7.1.1, and the series $\varpi(s)$ may be proven tame in Proposition 7.12. It will be possible to shift the contour of the Rice integral to the left, use the Rice method and apply Proposition 7.10. This leads in Theorem 7.11 to the asymptotics of $QM_\pm(n)$ and $QV_\alpha(n)$, from which we derive the asymptotics of **QuickRand**.
- (b) For the $\text{QuickQuant}_\alpha(n)$ algorithm, we operate in an indirect way. We *do not use* the general scheme, and we deal with $QQ_\alpha(n)$ by computing $QV_\alpha(n)$, with the help of the relation (7.13). This is done in Section 7.5; with a strong use of the Laplace method.

7.4 Asymptotic Analysis of QuickVal

7.4.1 Analytic lifting and Rice formula.

We deal with the analytic lifting $s \mapsto \varpi(s)$ of coefficients $\varphi(k)$ computed in Lemma 7.6. Here, for each algorithm **QuickMin**, **QuickMax** or QuickVal_α , the analytic lifting exists in the domain $\Re s > 1$.

Lemma 7.9. *For each algorithm **QuickMin**, **QuickMax** and QuickVal_α , the mixed Dirichlet series are equal to*

$$\varpi_-(s) = 2\mathcal{J}[t^{s-2}], \quad \varpi_+(s) = 2\mathcal{J}[(1-u)^{s-2}] \quad \varpi(\alpha, s) = \frac{4}{s+1} \mathcal{J}[(t \vee \alpha - u \wedge \alpha)^{s-2}].$$

We wish to apply Rice Method, and we need the series $\varpi(s)$ to be tame at $\sigma_0 = 1$. This will be the case, as we will prove in Proposition 7.12. We will also prove in the same Proposition 7.12 that the Dirichlet series $\varpi(s)$ remains analytic at $s = 1$. (This means in our vocabulary that it is tame at $s = 1$ with order 0). Then, Rice method leads in this case to the following result:

Proposition 7.10. *If the mixed Dirichlet $\varpi(s)$ is tame at $s = \sigma_0 = 1$ with order 0 and width δ , then the following asymptotics holds for the mean number of symbol comparisons,*

$$S(n) = \varpi(1) \cdot n + O(n^{1-\delta'}), \quad \text{with } \delta' < \delta.$$

7.4.2 Asymptotic estimates for the mean number of symbol comparisons for QuickMin, QuickMax, QuickVal and QuickRand.

Then, if we prove the series $\varpi(s)$ to be tame at $s = 1$ with order 0, we apply Proposition 7.10 and obtain the first asymptotic estimates which entail Assertions (b) and (c) of Theorem 7.2.

Theorem 7.11. *The following holds:*

- (a) *Consider a Π -tame source with width $\delta_0 < 1/2$. Then, the mean number of symbol comparisons for the **QuickVal** $_\alpha(n)$ algorithm satisfies*

$$QV_\alpha(n) = n\rho(\alpha) + O(n^{1-\delta}), \quad \text{with } \delta < 1 - 2\delta_0.$$

*The constant $\rho(\alpha)$ is the value of the mixed series $\varpi(\alpha, s)$ at $s = 1$, and the constant hidden in the O -term is uniform for $\alpha \in [0, 1]$. The function $\alpha \mapsto \rho(\alpha)$ is called the asymptotic slope of the **QuickVal** algorithm.*

- (b) *Consider a Π -tame source with width $\delta_0 < 1/2$. Then, the mean number of symbol comparisons for the **QuickRand** (n) algorithm satisfies*

$$QR(n) = n \left[\int_{\mathcal{I}} \rho(\alpha) d\alpha \right] + O(n^{1-\delta}), \quad \text{with } \delta < 1 - 2\delta_0.$$

- (c) *Consider a weakly Λ -tame source with width $\delta_1 < 1/2$. Then, the mean number of symbol comparisons for the **QuickMin** (n) and **QuickMax** (n) algorithm satisfy*

$$QM_\pm(n) = n\rho_\pm + O(n^{1-\delta}), \quad \text{with } \delta < 1 - 2\delta_1.$$

The constant ρ_\pm is the value of the mixed series $\varpi_\pm(s)$ at $s = 1$. The equalities $\rho_- = \rho(0)$ and $\rho_+ = \rho(1)$ hold.

Proof. This is a clear application of the Rice formula, as stated in Proposition 7.10. We will see that the upper bounds obtained in Lemmas 7.13 and 7.14 do not depend on $\alpha \in [0, 1]$. Thus the hidden constant in the O -term does not depend on α . \square

It then remains to study the tameness of $\varpi(s)$ at $s = 1$.

7.4.3 Tameness of the mixed series $\varpi(s)$.

We have already mentioned in Chapter 5.4.2, and more precisely describe in Section 7.1.1 two types of properties of the source that may entail tameness for the mixed series $\varpi(s)$. They are related to the convergence abscissae of the two Dirichlet series: see Definition 7.1 and Section 3.3 of Chapter 3.

We will show the following result, which relates the tameness of the mixed series $\varpi(\alpha, s)$ to the tameness of the source.

Proposition 7.12. *Consider a source, and for $\alpha \in [0, 1]$, its mixed series $\varpi_-(s)$, $\varpi_+(s)$ relative to **QuickMin**, **QuickMax**, and the series $\varpi(\alpha, s)$ relative to the **QuickVal** $_\alpha$ algorithm. Then, the following holds:*

- (a) *If the source is weakly Λ -tame with width $\delta_1 < 1/2$, then the mixed series $\varpi_-(s)$, $\varpi_+(s)$, $\varpi(0, s)$, $\varpi(1, s)$ are tame at $s = 1$ with width $\delta \leq 1 - 2\delta_1$ and order 0.*
- (b) *If the source is Π -tame with width $\delta_0 < 1/2$, then the mixed series $\varpi(\alpha, s)$ for any $\alpha \in]0, 1[$ are tame at $s = 1$ with width $\delta \leq 1 - 2\delta_0$ and order 0.*

We recall the relation between Π -tameness and weakly Λ -tameness: If the source is Π -tame with width δ_0 , it is also weakly Λ -tame with width δ_0 . But there are instances (for instance intermittent sources) for which the inverse implication is not true.

7.4.4 Principles of the proof of Proposition 7.12.

We recall that the analysis of the **QuickMin** or **QuickMax** algorithms is based on the study of the mixed series

$$\varpi_{\pm}(s) = 2\mathcal{J}[\phi_{\pm}^{s-2}], \quad \text{with } \phi_{-}(u, t) = t, \quad \phi_{+}(u, t) = 1 - u,$$

whereas the analysis of the algorithm **QuickVal** $_{\alpha}$ is based on the study of the mixed series

$$\varpi(\alpha, s) := \frac{4}{s+1} \underline{\varpi}(\alpha, s) \quad \text{with } \underline{\varpi}(\alpha, s) := \mathcal{J}[\phi_{\alpha}^{s-2}] = \sum_{w \in \Sigma^*} \int_{\mathcal{T}_w} \phi_{\alpha}(u, t)^{s-2} du dt, \quad (7.17)$$

which involves the function

$$\phi_{\alpha}(u, t) := t \vee \alpha - u \wedge \alpha. \quad (7.18)$$

For $\alpha = 0$, the function $\phi_0(u, t)$ coincides with the function $\phi_{-}(u, t) = t$ of **QuickMin** on the whole triangle \mathcal{T} , whereas, for $\alpha = 1$, the function $\phi_1(u, t)$ coincides with the function $\phi_{+}(u, t) = 1 - u$ of **QuickMax** on the whole triangle \mathcal{T} . And, for $\alpha \in]0, 1[$, there are three domains for the function ϕ_{α} , shown in Figure 7.3:

- (i) The rectangle $\mathcal{R} = [0, \alpha] \times [\alpha, 0]$ where $\phi_{\alpha}(u, t)$ equals $t - u$.
- (ii) The lower triangle $\mathcal{T}_{-} = \mathcal{T}_{[0, \alpha]}$, with basis $[0, \alpha]$, where $\phi_{\alpha}(u, t)$ equals $\alpha - u$.
- (iii) The upper triangle $\mathcal{T}_{+} = \mathcal{T}_{[\alpha, 1]}$, with basis $[\alpha, 1]$, where $\phi_{\alpha}(u, t)$ equals $t - \alpha$.

There is only one domain of interest (the whole triangle \mathcal{T}) when α equals 0 or 1.

$$\varpi_{-}(s) = \frac{s+1}{2} \varpi(0, s), \quad \varpi_{+}(s) = \frac{s+1}{2} \varpi(1, s).$$

The important behavior of the mixed series $\varpi(s)$ is around $\Re s = 1$ and more precisely at $s = 1$, where the factor $(s+1)/2$ equals 1. This explains why the behaviours of **QuickMin** and **QuickVal** $_0$ at one hand, and the behaviours of **QuickMax** and **QuickVal** $_1$ on the other hand, are quite similar.

Scheme of the proof of Proposition 7.12. For a domain $\mathcal{U} \subset \mathcal{T}$ and any integer $\ell \geq 2$, and any function $\phi \in L^1(\mathcal{T})$ we define

$$\mathcal{J}^{(\mathcal{U})}[\phi] := \int_{\mathcal{U}} (\gamma(u, t) + 1) \phi(u, t) du dt, \quad \mathcal{J}_{\ell}^{(\mathcal{U})}[\phi] := \int_{\mathcal{T}^{(\ell)} \cap \mathcal{U}} \phi(u, t) du dt,$$

where $\mathcal{T}^{(\ell)}$ is the union of all triangles \mathcal{T}_w with $w \in \Sigma^{\ell}$. Then the following decomposition holds

$$\mathcal{J}^{(\mathcal{U})}[\phi] = \sum_{\ell \geq 0} \mathcal{J}_{\ell}^{(\mathcal{U})}[\phi].$$

In the case of a general α , the integral $\varpi(\alpha, s)$ is the sum of three terms. Each term is of the form

$$\varpi^{(\mathcal{U})}(\alpha, s) = \frac{4}{s+1} \underline{\varpi}^{(\mathcal{U})}(\alpha, s), \quad \text{with } \underline{\varpi}^{(\mathcal{U})}(\alpha, s) := \mathcal{J}^{(\mathcal{U})}[\phi_{\alpha}^{s-2}],$$

and \mathcal{U} is any subset \mathcal{R}_{α} , $\mathcal{T}_{[0, \alpha]}$ or $\mathcal{T}_{[\alpha, 1]}$. Furthermore, the following decomposition holds

$$\underline{\varpi}^{(\mathcal{U})}(\alpha, s) = \sum_{\ell \geq 0} \underline{\varpi}_{\ell}^{(\mathcal{U})}(\alpha, s), \quad \text{with } \underline{\varpi}_{\ell}^{(\mathcal{U})}(\alpha, s) := \mathcal{J}_{\ell}^{(\mathcal{U})}[\phi_{\alpha}^{s-2}].$$

We first consider the integral on the two triangles, then on the (possible) rectangle.

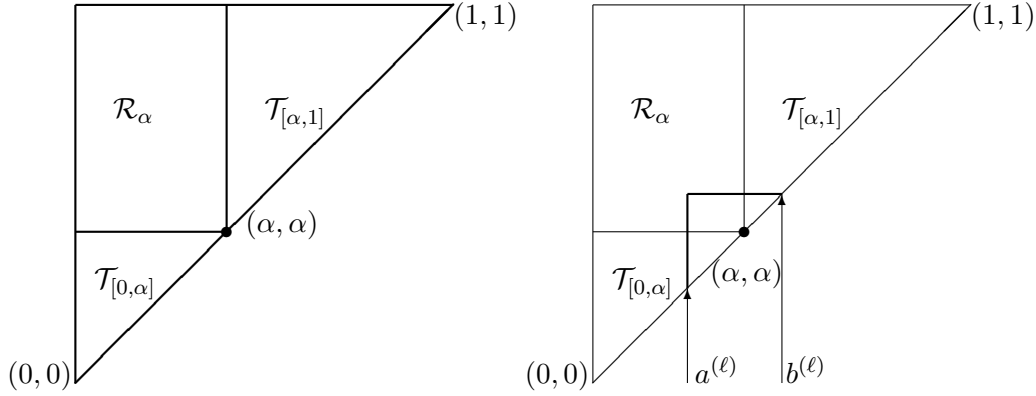


Figure 7.3: The three domains useful for the study of $\varpi(\alpha, s)$ (left) and the straddling triangle (right).

7.4.5 Study of integrals over triangles.

We first consider integrals over triangles $\mathcal{T}_{[0,\alpha]}$, $\mathcal{T}_{[\alpha,1]}$ and we prove the following estimates:

Lemma 7.13. *For $\mathcal{U} \in \{\mathcal{T}_{[0,\alpha]}, \mathcal{T}_{[\alpha,1]}\}$, the function $\varpi^{(\mathcal{U})}(\alpha, s)$ satisfies the inequality, for $\sigma := \Re s$,*

$$\left| \varpi^{(\mathcal{U})}(\alpha, s) \right| \leq C(\sigma) \Lambda \left(1 + \frac{\sigma}{2}\right) \leq C(\sigma) \Pi \left(\frac{\sigma}{2}\right).$$

for some constant $C(\sigma)$ uniformly bounded for $\sigma \geq \sigma_2 > 0$ (for any $\sigma_2 > 0$). If the source is weakly Λ -tame with width $\delta_1 < 1/2$, then the function $\varpi^{(\mathcal{U})}(\alpha, s)$ is tame at $\sigma = 1$, with width $\delta \leq 1 - 2\delta_1$ and order 0.

Proof. This proof is an easy extension of the proof which is done in [16] for the QuickMin algorithm.

Note that the source \mathcal{S} and its symmetric source $\check{\mathcal{S}}$ described in Section 2.4.2 have, for each k , the same set $\{p_w : w \in \Sigma^k\}$ of fundamental probabilities for prefixes of length k , and thus have the same Dirichlet series Λ and Π .

It is sufficient to deal with one of the two cases for \mathcal{U} . Indeed, the mixed Dirichlet series $\varpi(\alpha, s)$ relative to a source \mathcal{S} and the mixed m Dirichlet series $\check{\varpi}(\alpha, s)$ relative to the symmetric source $\check{\mathcal{S}}$ satisfy the identity

$$\check{\varpi}^{(\mathcal{T}_{[0,\alpha]})}(\alpha, s) = \varpi^{(\mathcal{T}_{[1-\alpha,1]})}(1 - \alpha, s).$$

Since the two sources have the same Dirichlet series $\Pi(s) := \check{\Pi}(s)$, they share the same tameness properties.

We consider the function on the upper triangle, and we study the “underlined” functions, namely

$$\underline{\varpi}^{(\mathcal{T}_{[\alpha,1]})}(\alpha, s) = \sum_{\ell \geq 0} \underline{\varpi}_\ell^{(\mathcal{T}_{[\alpha,1]})}(\alpha, s), \quad \underline{\varpi}_\ell^{(\mathcal{T}_{[\alpha,1]})}(\alpha, s) = \sum_{w \in \Sigma^\ell} \int_{\mathcal{T}_w \cap \mathcal{T}_{[\alpha,1]}} (t - \alpha)^{s-2} du dt.$$

For each ℓ , we consider some real $A \in [\alpha, 1]$ (to be fixed later as a function of σ and ℓ) and split the sum into three sums, each of them relative to a subset of prefixes: the prefixes w for which the right end b_w belongs to $[\alpha, A[$, the prefixes w for which the left end a_w belongs to $[A, 1]$ and finally the unique prefix τ for which $A \in [a_\tau, b_\tau]$. We omit the reference to the real α , and the three sums are respectively denoted by $\underline{\varpi}_\ell^{(+)}(\sigma)$, $\underline{\varpi}_\ell^{(-)}(\sigma)$, $\underline{\varpi}_\ell^{(=)}(\sigma)$.

For $b_w \in [\alpha, A]$, we use the inequality

$$\int_{a_w}^{b_w} (t - a_w)(t - \alpha)^{\sigma-2} dt \leq \int_{a_w}^{b_w} (t - \alpha)^{\sigma-1} dt,$$

yielding

$$\underline{\varpi}_\ell^{(-)}(\sigma) \leq \int_{\alpha}^A (t - \alpha)^{\sigma-1} dt = \frac{1}{\sigma} (A - \alpha)^{\sigma}.$$

For $a_w \in [A, 1]$, we observe that

$$\int_{a_w}^{b_w} (t - a_w)(t - \alpha)^{\sigma-2} dt \leq \frac{1}{2} (A - \alpha)^{\sigma-2} p_w^2, \quad \text{so that} \quad \underline{\varpi}_\ell^{(+)}(\sigma) \leq \frac{1}{2} (A - \alpha)^{\sigma-2} \Lambda_\ell^{[\alpha,1]}(2).$$

We now choose A such that the two previous bounds are equal, namely

$$A - \alpha = \left(\frac{\sigma}{2} \Lambda_\ell^{[\alpha,1]}(2) \right)^{1/2} \leq (1 - \alpha),$$

so that A belongs to the interval $[\alpha, 1]$. Then

$$\underline{\varpi}_\ell^{(+)}(\sigma) + \underline{\varpi}_\ell^{(-)}(\sigma) \leq C_1(\sigma) \left[\Lambda_\ell^{[\alpha,1]}(2) \right]^{\sigma/2},$$

where $C_1(\sigma)$ is bounded for $\sigma \geq \sigma_2$ (for any $\sigma_2 > 0$). The middle part $\underline{\varpi}_\ell^{(=)}(\sigma)$ corresponds to the fundamental interval $[a_\tau, b_\tau]$ of length $p_\tau \leq \Lambda_\ell^{[\alpha,1]}(2)^{1/2}$, and

$$\underline{\varpi}_\ell^{(=)}(\sigma) \leq \int_{a_\tau}^{b_\tau} (t - \alpha)^{\sigma-1} dt \leq \frac{1}{\sigma} (A - \alpha + p_\tau)^{\sigma} \leq C_2(\sigma) \left[\Lambda_\ell^{[\alpha,1]}(2) \right]^{\sigma/2},$$

where $C_2(\sigma)$ is bounded for $\sigma \geq \sigma_2$ (for any $\sigma_2 > 0$). Finally, we use the inequality $\Lambda_\ell^{[\alpha,1]}(2) \leq \Lambda_\ell(2)$ and the log-convexity of $\Lambda_\ell(2)$, which entails the inequality for $\sigma \leq 2$

$$\Lambda_\ell(2)^{\sigma/2} = \Lambda_\ell(2)^{\sigma/2} \Lambda_\ell(1)^{1-\sigma/2} \leq \Lambda_\ell(2^{\frac{\sigma}{2}} + 1 - \frac{\sigma}{2}) = \Lambda_\ell(1 + \frac{\sigma}{2}),$$

and, taking the sum over integers ℓ , we have obtained the bound for the integral over the upper triangle,

$$\varpi^{(\mathcal{T}_{[1,\alpha]})}(\alpha, s) \leq C(\sigma) \Lambda_\ell(1 + \frac{\sigma}{2}) \leq C(\sigma) \Pi\left(\frac{\sigma}{2}\right).$$

This is the same bound for the integral on the lower triangle. \square

7.4.6 Study of the integral over the rectangle \mathcal{R} .

We prove the following:

Lemma 7.14. *The integral $\varpi^{(\mathcal{R})}(\alpha, s)$ satisfies*

$$\left| \varpi^{(\mathcal{R})}(\alpha, s) \right| \leq C(\sigma) \Pi(\sigma) \quad \text{for any } \sigma > 0.$$

If the source is Π -tame with width δ_0 , the mapping $s \mapsto \varpi^{(\mathcal{R})}(\alpha, s)$ is tame at $\sigma = 1$ with width $\delta \leq 1 - \delta_0$ and order 0.

Proof. The integral $\underline{\varpi}^{(\mathcal{R})}(\alpha, s)$ is the sum of the integrals $\underline{\varpi}_\ell^{(\mathcal{R})}(\alpha, s)$ defined as

$$\underline{\varpi}_\ell^{(\mathcal{R})}(\alpha, s) := \int_{\mathcal{T}^{(\ell)} \cap \mathcal{R}} (t - u)^{s-2} du dt.$$

For each depth ℓ , and for each α , there exists a unique word $w^{(\ell)}$ whose fundamental interval of the form $[a^{(\ell)}, b^{(\ell)}]$ contains α . Then, the intersection $\mathcal{T}^{(\ell)} \cap \mathcal{R}$ is the rectangle $[a^{(\ell)}, \alpha] \times [\alpha, b^{(\ell)}]$, and

$$\underline{\varpi}_\ell^{(\mathcal{R})}(\alpha, \sigma) = \int_\alpha^{b^{(\ell)}} dt \int_{a^{(\ell)}}^\alpha (t - u)^{\sigma-2} du = \frac{1}{\sigma(\sigma-1)} \left[(b^{(\ell)} - a^{(\ell)})^\sigma - (b^{(\ell)} - \alpha)^\sigma - (\alpha - a^{(\ell)})^\sigma \right].$$

If we let

$$c = \frac{b^{(\ell)} - \alpha}{b^{(\ell)} - a^{(\ell)}}, \quad d = \frac{\alpha - a^{(\ell)}}{b^{(\ell)} - a^{(\ell)}}, \quad \text{with } c + d = 1,$$

then the integral is written as

$$\underline{\varpi}_\ell^{(\mathcal{R})}(\alpha, \sigma) = \frac{1}{\sigma(\sigma-1)} (b^{(\ell)} - a^{(\ell)})^\sigma \left[c(1 - c^{\sigma-1}) + d(1 - d^{\sigma-1}) \right].$$

There are two cases for $\sigma := \Re s$. Consider any $\rho_0 > 0$. The (easy) first case arises for $\sigma \geq 1 + \rho_0 > 1$, where

$$\underline{\varpi}_\ell^{(\mathcal{R})}(\sigma) \leq \frac{2}{\rho_0(1 + \rho_0)} (b^{(\ell)} - a^{(\ell)})^\sigma \leq \frac{2}{\rho_0(1 + \rho_0)} \Pi_\ell(\sigma).$$

Consider now the case when $|\sigma - 1| \leq \rho_0$, and let $\rho := \sigma - 1$. The series expansion of $x \mapsto x^\rho$

$$x^\rho - 1 = \sum_{k \geq 1} \frac{\rho^k}{k!} \log^k x = \rho \log x + \rho^2 |\log x|^2 \sum_{k \geq 0} \frac{\rho^k}{(k+2)!} \log^k x$$

provides the estimate

$$|x^\rho - 1 - \rho \log x| \leq \rho^2 |\log x|^2 e^{|\rho \log x|}.$$

For $x \in [0, 1]$, one has $|\log x| = -\log x$ and $e^{|\rho| |\log x|} = e^{-|\rho| \log x} = x^{-|\rho|}$, so that

$$\left| \frac{x(1 - x^\rho)}{\rho} + x \log x \right| \leq \rho x \log^2 x e^{|\rho| |\log x|} = \rho x^{1-|\rho|} \log^2 x.$$

If now $|\rho| \leq \rho_0$, with $\rho_0 < 1$, there exists C_2 (which depends on ρ_0) for which the inequality $x^{1-|\rho|} \log^2 x \leq C$ holds for any $x \in [0, 1]$ and any ρ with $|\rho| \leq \rho_0$. Furthermore, the entropic sum $h(x) = -x \log x - (1 - x) \log(1 - x)$ belongs to $[0, \log 2]$, and finally

$$\underline{\varpi}_\ell^{(\mathcal{R})}(\alpha, \sigma) = \frac{2}{\sigma} (b^{(\ell)} - a^{(\ell)})^\sigma [h(c) + |\sigma - 1| O(1)].$$

Returning to $\varpi(\alpha, s)$ itself and taking the sum over all the integers ℓ gives the final bound

$$\varpi^{(\mathcal{R})}(\alpha, \sigma) \leq C(\sigma) \Pi(\sigma), \quad \text{for any } \sigma \geq \sigma_0 > 0.$$

Now, the source is Π -tame with width δ_0 , the series $\Pi(\sigma)$ is convergent on the domain $\Re s > \delta_0$. \square

7.4.7 End of the proof.

With the two Lemmas, and the formula,

$$\varpi(\alpha, s) = \varpi^{(\mathcal{T}_-)}(\alpha, s) + \varpi^{(\mathcal{T}_+)}(\alpha, s) + \varpi^{(\mathcal{R})}(\alpha, s),$$

we obtain the result for QuickVal_α. Observe that, in the cases $\alpha = 0$ or $\alpha = 1$, there is only one triangle (either \mathcal{T}_- or \mathcal{T}_+) and no rectangle, and the weak Λ -tameness is then sufficient to conclude.

7.5 Return to the analysis of the QuickQuant_α Algorithm.

7.5.1 Regularity of the asymptotic slope $\alpha \mapsto \rho(\alpha)$ of QuickVal.

Let us first recall that the function $\varphi_\alpha(u, t)$ is defined in Lemma 7.6 as

$$\varphi_\alpha(k) = \frac{4}{k+1} \mathcal{J} \left[(t \vee \alpha - u \wedge \alpha)^{k-2} \right],$$

with the notation $x \vee y = \max(x, y)$ and $x \wedge y = \min(x, y)$. We note as in Eq. (7.18)

$$\phi_\alpha(u, t) := t \vee \alpha - u \wedge \alpha.$$

The asymptotic slope $\rho(\alpha)$ for the mean number of symbol-comparisons is given by

$$\rho(\alpha) = 2\mathcal{J}[\phi_\alpha^{-1}] = \int_{\mathcal{T}} [\gamma(u, t) + 1] \phi_\alpha(u, t)^{-1} du dt.$$

in order to compare the behaviors of the two algorithms QuickVal and QuickQuant, we need the asymptotic slope to be regular enough.

Lemma 7.15. *For a source which is Π -tame of width $\delta_0 < 1/2$, the asymptotic slope $\alpha \mapsto \rho(\alpha)$ of the QuickVal algorithm is Hölder with exponent η with $\eta < 1 - 2\delta_0$: for any η with $0 < \eta < 1 - 2\delta_0$, there exists a constant C_η for which*

$$|\rho(\beta) - \rho(\alpha)| \leq C_\eta |\beta - \alpha|^\eta \quad \forall (\alpha, \beta) \in [0, 1]^2.$$

Proof. We assume that the inequality $\alpha \leq \beta$ holds and begin with the inequality

$$|\rho(\beta) - \rho(\alpha)| \leq 2\mathcal{J} \left[\left| \phi_\alpha^{-1} - \phi_\beta^{-1} \right| \right].$$

There are six domains, described in Figure 7.4 and listed in the first column of Figure 7.5. On each domain, we obtain an upper bound for the function $\left| \phi_\alpha^{-1} - \phi_\beta^{-1} \right|$, of the form

$$\left| \phi_\alpha^{-1} - \phi_\beta^{-1} \right| \leq |\beta - \alpha|^\eta \left(\phi_\alpha^{-1-\eta} + \phi_\beta^{-1-\eta} \right), \quad \text{for any } \eta \in [0, 1].$$

We do not directly obtain such an upper bound: on each domain we get a “natural” upper bound of the form

$$\left| \phi_\alpha^{-1} - \phi_\beta^{-1} \right| \leq |\beta - \alpha|^\eta F_{\alpha, \beta}, \tag{7.19}$$

and on each domain the function $F_{\alpha, \beta}$ is related to the functions ϕ_α or ϕ_β or both. This approach is summarized in Figure 7.5. The exponent η will be chosen later on.

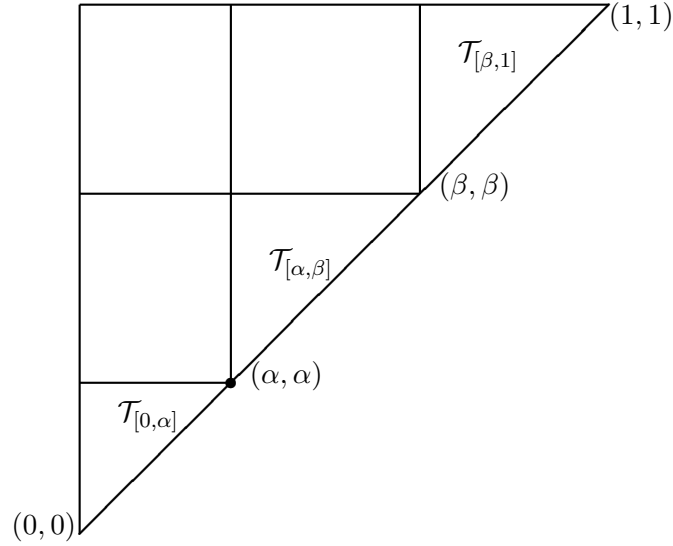


Figure 7.4: The six domains for the study of $\rho(\alpha) - \rho(\beta)$ for any $\eta \in [0, 1]$.

Domain \mathcal{D}	$F_{\alpha,\beta}(u, t)$ for $(u, t) \in \mathcal{D}$
Triangle $\mathcal{T}_{[0,\alpha]}$	$(\alpha - u)^{-1-\eta} = \phi_\alpha^{-1-\eta}(u, t)$
Triangle $\mathcal{T}_{[\beta,1]}$	$(t - \beta)^{-1-\eta} = \phi_\beta^{-1-\eta}(u, t)$
Triangle $\mathcal{T}_{[\alpha,\beta]}$	$(t - \alpha)^{-1-\eta} + (\beta - u)^{-1-\eta} = \phi_\alpha^{-1-\eta}(u, t) + \phi_\beta^{-1-\eta}(u, t)$
Rectangle $[\alpha, \beta] \times [\beta, 1]$	$(t - u)^{-1-\eta} = \phi_\beta^{-1-\eta}(u, t)$
Rectangle $[0, \alpha] \times [\alpha, \beta]$	$(t - u)^{-1-\eta} = \phi_\alpha^{-1-\eta}(u, t)$
Rectangle $[0, \alpha] \times [\beta, 1]$	0

Figure 7.5: The six domains (in the first column). The second column describes the “natural” intermediary function $F_{\alpha,\beta}$ described in (7.19), that is expressed as a function of ϕ_α and ϕ_β .

We first explain how to obtain the results described in the tabular of Figure 7.5. Using the symmetric source, it is enough to study the function $|\phi_\alpha^{-1} - \phi_\beta^{-1}|$ on three domains: two triangles $\mathcal{T}_{[0,\alpha]}$ and $\mathcal{T}_{[\alpha,\beta]}$ and one rectangle $[\alpha, \beta] \times [\beta, 1]$.

On the triangle $\mathcal{T}_{[0,\alpha]}$, with the two inequalities $(\beta - \alpha) \leq (\beta - u)$ and $(\alpha - u) \leq (\beta - u)$, we have

$$\left| \phi_\beta^{-1}(u, t) - \phi_\alpha^{-1}(u, t) \right| = \frac{(\beta - \alpha)}{(\beta - u)(\alpha - u)} \leq \frac{(\beta - \alpha)^\eta}{(\beta - u)^\eta(\alpha - u)} \leq \frac{(\beta - \alpha)^\eta}{(\alpha - u)^{1+\eta}}.$$

On the rectangle $[\alpha, \beta] \times [\beta, 1]$, using the inequalities $(u - \alpha) \leq (\beta - \alpha) \leq (t - \alpha)$ and $(t - u) \leq (t - \alpha)$, we have

$$\left| \phi_\beta^{-1}(u, t) - \phi_\alpha^{-1}(u, t) \right| = \frac{(u - \alpha)}{(t - \alpha)(t - u)} \leq (\beta - \alpha)^\eta \frac{(t - \alpha)^{1-\eta}}{(t - \alpha)(t - u)} \leq \frac{(\beta - \alpha)^\eta}{(t - u)^{1+\eta}}.$$

On the triangle $\mathcal{T}_{[\alpha, \beta]}$, with the inequalities $\beta - \alpha \geq t - \alpha$ and $\beta - \alpha \geq \beta - u$, we obtain

$$\left| \phi_{\beta}^{-1}(u, t) - \phi_{\alpha}^{-1}(u, t) \right| \leq \frac{1}{t - \alpha} + \frac{1}{\beta - u} \leq (\beta - \alpha)^{\eta} \left(\frac{1}{(t - \alpha)^{1+\eta}} + \frac{1}{(\beta - u)^{1+\eta}} \right).$$

Now, with the results of Figure 7.5, and, on each of the six domains \mathcal{U} described in the first column of Figure 7.5, we use the inequality

$$\mathcal{J}^{(\mathcal{U})} \left[\left| \phi_{\alpha}^{-1} - \phi_{\beta}^{-1} \right| \right] \leq |\beta - \alpha|^{\eta} \left(\mathcal{J}^{(\mathcal{U})} \left[|\phi_{\alpha}|^{-1-\eta} \right] + \mathcal{J}^{(\mathcal{U})} \left[|\phi_{\beta}|^{-1-\eta} \right] \right).$$

We take the sum over the six domains \mathcal{U} , and we obtain

$$\mathcal{J} \left[|\phi_{\alpha}|^{-1} - |\phi_{\beta}|^{-1} \right] \leq |\beta - \alpha|^{\eta} (\varpi(\alpha, 1 - \eta) + \varpi(\beta, 1 - \eta)).$$

We now use the tameness of the function $\varpi(\alpha, s)$ at $s = 1$ with a width $\delta \leq 1 - 2\delta_0$, with the the uniform bounds that are exhibited in Proposition 7.12. Then, choosing any η with $0 < \eta < \delta$, we obtain, with $C_{\eta} := 2 \sup \{ \varpi(\alpha, 1 - \eta), \alpha \in [0, 1] \}$,

$$\mathcal{J} \left[\left| \phi_{\alpha}^{-1} - \phi_{\beta}^{-1} \right| \right] \leq |\beta - \alpha|^{\eta} [\varpi(\alpha, 1 - \eta) + \varpi(\beta, 1 - \eta)] \leq C_{\eta} |\beta - \alpha|^{\eta}.$$

□

7.5.2 Return to the analysis of QuickQuant.

We return to the analysis of the QuickQuant algorithm with Eq. (7.13) and will prove the following result that exactly entails Assertion (a) of Theorem 7.11.

Theorem 7.16. *Consider a source, assumed to be Π -tame with width $\delta_0 < 1/2$. Then, for any $\alpha \in [0, 1]$, the mean number of symbols comparisons performed by the QuickQuant $_{\alpha}$ algorithm satisfies,*

$$\mathcal{Q}\mathcal{Q}_{\alpha}(n) = n\rho(\alpha) + O(n^{1-\kappa/2}),$$

with any κ with $0 < \kappa < 1 - 2\delta_0$. Moreover, the constant hidden in the O -term is uniform on any interval of the form $[\alpha_1, 1 - \alpha_1]$ with $\alpha_1 > 0$.

We first remark that the theorem is already proven for $\alpha = 0$ or $\alpha = 1$. Indeed, for $\alpha = 0$, the QuickQuant $_{\alpha}$ algorithm coincides with QuickMin and for $\alpha = 1$, with QuickMax. In these cases, this is just an application of Theorem 7.11.

We now focus on the case when α belongs to the open interval $]0, 1[$. The form of Eq. (7.13) together with the estimate for $\mathcal{Q}\mathcal{V}_{\alpha}(n)$ obtained in Theorem 7.11 leads us to the following estimate for $\mathcal{Q}\mathcal{Q}_{\alpha}(n)$

$$\mathcal{Q}\mathcal{Q}_{\alpha}(n) = \int_0^1 f_n(\alpha, v) \mathcal{Q}\mathcal{V}_v(n) dv = \int_0^1 f_n(\alpha, v) \left[n\rho(v) + O(n^{1-\delta}) \right] dv.$$

Since the constant hidden in the O -constant is uniform with respect to v , we obtain

$$\mathcal{Q}\mathcal{Q}_{\alpha}(n) = n \cdot I_n(\alpha) + O(n^{1-\delta}) \quad \text{with} \quad I_n(\alpha) := \int_0^1 f_n(\alpha, v) \rho(v) dv. \quad (7.20)$$

We wish to study the integral $I_n(\alpha)$, and compare it to the value $\rho(\alpha)$. As the density $f_n(\alpha, v)$ can be compared to a large power of a function f_{α} , we are then led to apply the Laplace method that is now recalled.

7.5.3 Laplace's method.

Laplace's method is a method that is widely used to approximate the integral of a large power of a given function f

$$I_n := \int_a^b f(v)^n dv,$$

where f is a function of class \mathcal{C}^2 , n is a large integer, and the end points a and b are possibly infinite. Assume that the function $|f(v)|$ has a unique strict maximum at v_0 in the interior of the interval $[a, b]$, with $f(v_0)$ and $f''(v_0)$ non zero. Then, the following estimate holds

$$I_n := \int_a^b f(v)^n dv \sim_{n \rightarrow \infty} \frac{1}{\sqrt{n}} f(v_0)^n \sqrt{-\frac{2\pi f(v_0)}{f''(v_0)}}.$$

Figure 7.6 gives an explanation of the estimate.

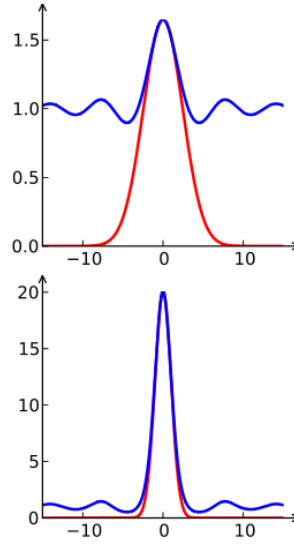


Figure 7.6: Here, we consider the function f^n , where f is defined on the interval $[-20, 20]$ and equals $f(v) = \exp[\sin v/v]$. The function f has a strict maximum at $v_0 = 0$. On the top, we see the function f^n for $n = 1/2$ in blue, and, at the bottom, for $n = 3$. When n becomes large, the approximation of this function by the Gaussian function (shown in red) is getting better. This observation underlies Laplace's method.

We now give an idea of the proof: The Taylor expansion of $g := \log f$ at v_0 gives the estimate

$$g(v) = g(v_0) + (v - v_0)g'(v_0) - \frac{\lambda}{2}(v - v_0)^2,$$

with

$$g'(v_0) = \frac{f'(v_0)}{f(v_0)}, \quad \lambda := -g''(v_0) = \frac{f'^2(v_0) - f''(v_0)f(v_0)}{f(v_0)^2}.$$

As v_0 is a strict maximum for f , the function g satisfies at v_0 , the equalities $g'(v_0) = 0$ and $\lambda = -f''(v_0)/f(v_0) > 0$ hold and entail the following estimate

$$g(v) \sim g(v_0) - \frac{\lambda}{2}(v - v_0)^2.$$

When n becomes large, the integral can be compared to the integral of the Gaussian function $x \mapsto \exp[-x^2/2]$ on the interval $[-\infty, +\infty[$,

$$\begin{aligned} \int_a^b \exp[ng(v)] dv &\sim \exp[ng(v_0)] \int_a^b \exp\left[-\frac{\lambda n}{2}(v - v_0)^2\right] dv \\ &\sim \exp[ng(v_0)] \int_{-\infty}^{+\infty} \exp\left[-\frac{\lambda n}{2}(v - v_0)^2\right] dv \\ &= \exp[ng(v_0)] \sqrt{\frac{2\pi}{\lambda n}} \end{aligned}$$

and the final estimate holds.

7.5.4 A tailored version of the Laplace method.

Since the function ρ is certainly not of class \mathcal{C}^2 , and we can only guarantee it is Hölder, we need a version of the Laplace method suited to our need.

Theorem (Laplace method). *Let ρ and f be positive functions defined over the unit interval $[0, 1]$ of the real line, and consider the integrals*

$$J_n := \int_0^1 \rho(v) f(v)^n dv, \quad K_n := \int_0^1 f(v)^n dv.$$

Assume the following:

- (i) ρ is Hölder with exponent η ;
- (ii) f is continuous on $[0, 1]$ and indefinitely derivable on $]0, 1[$;
- (iii) f attains its maximum at a unique point $v_0 \in]0, 1[$;
- (iv) The three real numbers $\rho(v_0)$, $f(v_0)$, $f''(v_0)$ are non zero.

Then, the integrals J_n and K_n satisfy

$$\frac{J_n}{K_n} = \rho(v_0) [1 + O(n^{-\eta/2})], \quad \text{with } \eta' < \eta.$$

Sketch of proof. In the Laplace method, the contribution of a small interval around v_0 gives the main asymptotic term. More precisely, one considers the two integrals

$$J_n^{(1)} := \int_{A_n} \rho(v) f(v)^n dv, \quad K_n^{(1)} := \int_{A_n} f(v)^n dv, \quad \text{with } A_n := \left[v_0 - \frac{\log n}{\sqrt{n}}, v_0 + \frac{\log n}{\sqrt{n}} \right].$$

First, the integral $K_n^{(1)}$ is proven to give the main term for K_n in the Laplace method, and, moreover, the Hölder hypothesis for ρ is enough to prove that $J_n^{(1)}$ gives the main term for J_n in the Laplace method. More precisely, one has

$$\frac{J_n^{(1)}}{J_n} = 1 + O\left(\frac{1}{n}\right), \quad \frac{K_n^{(1)}}{K_n} = 1 + O\left(\frac{1}{n}\right).$$

Second, on the interval A_n , the following estimate holds for ρ ,

$$\rho(v) = \rho(v_0) + O\left(\frac{\log n}{\sqrt{n}}\right)^\eta = \rho(v_0) [1 + O(n^{-\eta'/2})], \quad \text{with } \eta' < \eta,$$

and implies the estimates, since $\eta' < 1$,

$$J_n^{(1)} = \rho(v_0) \cdot K_n^{(1)} [1 + O(n^{-\eta'/2})], \quad \text{and thus } J_n = \rho(v_0) \cdot K_n [1 + O(n^{-\eta'/2})].$$

□

7.5.5 Proof of Theorem 7.16.

We now study the integral $I_n(\alpha)$ defined in (7.20). It involves the beta density $f_n(\alpha, v)$ of parameters $(\lfloor \alpha n + (1 - \alpha) \rfloor, n - \lfloor \alpha n + (1 - \alpha) \rfloor + 1)$ and we perform two steps. In the first step, we “forget” the integer parts in the expression of $f_n(\alpha, v)$.

Step 1. We deal with the Beta density $\widehat{f}_n(\alpha, v)$ of parameters $(\alpha n + 1 - \alpha, n - \alpha n + \alpha)$, which is written as

$$\widehat{f}_n(\alpha, v) = \widehat{C}(\alpha, n) f_\alpha(v)^{n-1},$$

with

$$\widehat{C}(\alpha, n) := \frac{\Gamma(n+1)}{\Gamma(\alpha n + 1 - \alpha) \Gamma(n - \alpha n + \alpha)}, \quad f_\alpha(v) := v^\alpha (1-v)^{(1-\alpha)}.$$

We wish to study the integral

$$\widehat{I}_n(\alpha) := \int_0^1 \rho(v) \widehat{f}_n(\alpha, v) dv.$$

We first deal with the integrals “without” the Gamma term $\widehat{C}(\alpha, n)$, namely

$$J_n(\alpha) := \int_I \rho(v) f_\alpha(v)^{n-1} dv, \quad K_n(\alpha) := \int_I f_\alpha(v)^{n-1} dv,$$

and use the Laplace method. We first check the hypotheses for the function f_α . For $\alpha \in]0, 1[$, the function $f_\alpha(v)$ satisfies

$$\begin{aligned} f'_\alpha(v) &= v^{\alpha-1} (1-v)^{-\alpha} (\alpha - v) & \text{so that } f'_\alpha(\alpha) &= 0, \\ f''_\alpha(\alpha) &= -\frac{1}{\alpha(1-\alpha)} f_\alpha(\alpha) & \text{so that } f''_\alpha(\alpha) &\neq 0, \end{aligned}$$

and $f_\alpha(v)$ attains its maximum at $v = \alpha$. Then, the Laplace method can be applied for any $\alpha \in]0, 1[$, and provides the estimate

$$J_n(\alpha) = \rho(\alpha) \cdot K_n(\alpha) \cdot [1 + O(n^{-\kappa'/2})].$$

Now, the initial integral of interest is

$$\widehat{I}_n(\alpha) := \int_0^1 \rho(v) \widehat{f}_n(\alpha, v) dv = \widehat{C}(\alpha, n) J_n(\alpha),$$

whereas the product $\widehat{C}(\alpha, n) K_n(\alpha)$ equals 1, since $\widehat{f}_n(\alpha, v)$ is a density. This leads to the estimate

$$\widehat{I}_n(\alpha) := \int_0^1 \rho(v) \widehat{f}_n(\alpha, v) dv = \rho(\alpha) \cdot [1 + O(n^{-\kappa'/2})].$$

Step 2. We now take into account the “integer part”, and we have to estimate the difference

$$\left| I_n(\alpha) - \widehat{I}_n(\alpha) \right| \leq \int_I \rho(v) \left| f_n(\alpha, v) - \widehat{f}_n(\alpha, v) \right| dv \leq K R_n \quad \text{with} \quad R_n := \int_I \left| f_n(\alpha, v) - \widehat{f}_n(\alpha, v) \right| dv,$$

and we compare R_n to the integrals

$$K_n(\alpha) := \int_0^1 f_n(\alpha, v) dv, \quad \widehat{K}_n(\alpha) := \int_0^1 \widehat{f}_n(\alpha, v) dv,$$

which are both equal to 1. We will show that R_n is “negligible” with respect to these integrals. Here, U_n is said to be negligible with respect to V_n if U_n/V_n is $O(1/n)$.

The two Beta densities $\widehat{f}_n(\alpha, v)$ and $f_n(\alpha, v)$ have the same shape: they are both unimodal, attain their maximum, respectively at $v = \alpha$ and $v = \lfloor \alpha n - \alpha \rfloor / n$, and are equal to 0 at $v = 0$ and $v = 1$. We write the difference $|f_n(\alpha, v) - \widehat{f}_n(\alpha, v)|$ as

$$|f_n(\alpha, v) - \widehat{f}_n(\alpha, v)| = |\widehat{f}_n(\alpha, v)| \left| 1 - \frac{f_n(\alpha, v)}{\widehat{f}_n(\alpha, v)} \right|,$$

and the ratio $f_n(\alpha, v)/\widehat{f}_n(\alpha, v)$ is a product of two ratios, the ratio between the Gamma terms, and the ratio between the functions. First, the ratio between the Gamma terms, namely

$$\frac{C(\alpha, n)}{\widehat{C}(\alpha, n)} = \frac{\Gamma(\alpha n + 1 - \alpha)\Gamma(n - \alpha n + \alpha)}{\Gamma(\lfloor \alpha n - \alpha \rfloor + 1)\Gamma(n - \lfloor \alpha n - \alpha \rfloor)},$$

is evaluated thanks to the Stirling formula for $x \rightarrow \infty$ and $a \in [0, 1]$,

$$\frac{\Gamma(x+a)}{\Gamma(x)} = x^a \left[1 + O\left(\frac{1}{x}\right) \right] \quad \text{and thus} \quad \frac{C(\alpha, n)}{\widehat{C}(\alpha, n)} = \left(\frac{\alpha}{1-\alpha} \right)^{\{\alpha n - \alpha\}} + O\left(\frac{1}{n}\right).$$

On any interval $[\alpha_1, 1 - \alpha_1]$, with $\alpha_1 > 0$, the ratio $\alpha/(1-\alpha)$ is bounded both from above and below, and the fractional part belongs to $[0, 1]$ so that

$$\frac{C(\alpha, n)}{\widehat{C}(\alpha, n)} = \left(\frac{\alpha}{1-\alpha} \right)^{\{\alpha n - \alpha\}} \left[1 + O\left(\frac{1}{n}\right) \right]. \quad (7.21)$$

Second, it is possible to choose an interval A_n for which the following holds:

- (i) The interval A_n contains the two points α and $\lfloor \alpha n - \alpha \rfloor / n = \alpha + O(1/n)$;
- (ii) The two integrals of $f_n(\alpha, v)$ and $\widehat{f}_n(\alpha, v)$ on the two intervals which form $I \setminus A_n$ are negligible with respect to the integral $K_n(\alpha)$ or $\widehat{K}_n(\alpha)$. It is then the same for the integral of the difference $|f_n(\alpha, v) - \widehat{f}_n(\alpha, v)|$;
- (iii) On the interval A_n , the ratio $(1-v)/v$ is uniformly bounded (with respect to n) both from below and above, and the ratio f_n/\widehat{f}_n satisfies

$$\begin{aligned} \frac{f_n(\alpha, v)}{\widehat{f}_n(\alpha, v)} &= \left(\frac{1-v}{v} \right)^{\{\alpha n - \alpha\}} \cdot \frac{C(\alpha, n)}{\widehat{C}(\alpha, n)} \\ &= \left(\frac{1-v}{v} \right)^{\{\alpha n - \alpha\}} \left(\frac{\alpha}{1-\alpha} \right)^{\{\alpha n - \alpha\}} \left[1 + O\left(\frac{1}{n}\right) \right]. \end{aligned}$$

These three properties for the interval A_n , entail the estimate, for $v \in A_n$,

$$\left| 1 - \frac{f_n(\alpha, v)}{\widehat{f}_n(\alpha, v)} \right| = g_n(v) + O\left(\frac{1}{n}\right), \quad \text{with} \quad g_n(v) = \left| 1 - \left(\frac{1-v}{v} \frac{\alpha}{1-\alpha} \right)^{\{\alpha n - \alpha\}} \right|,$$

and thus the function $|f_n(\alpha, v) - \widehat{f}_n(\alpha, v)|$ satisfies

$$\begin{aligned} |f_n(\alpha, v) - \widehat{f}_n(\alpha, v)| &= \widehat{f}_n(\alpha, v) \cdot \left| 1 - \frac{f_n(\alpha, v)}{\widehat{f}_n(\alpha, v)} \right| \\ &= \widehat{f}_n(\alpha, v) \left(O\left(\frac{1}{n}\right) + g_n(v) \right). \end{aligned}$$

We observe the upper bound

$$g_n(v) \leq g(v) \quad \text{with} \quad g(v) = \left| 1 - \left(\frac{1-v}{v} \frac{\alpha}{1-\alpha} \right) \right| = \left| \frac{v-\alpha}{v(1-\alpha)} \right|.$$

As the last function is zero at $v = \alpha$, the integral of $|f_n(\alpha, v) - \hat{f}_n(\alpha, v)|$ on the interval A_n is negligible with respect to the integral $\hat{K}_n(\alpha)$.

Conclusion. First, we recall that, for $\alpha = 0$, the **QuickQuant** $_\alpha$ algorithm coincides with **QuickMin**, and for $\alpha = 1$, the **QuickQuant** $_\alpha$ algorithm coincides with **QuickMax**. Then, the theorem holds for $\alpha = 0$ and $\alpha = 1$. For $\alpha \in]0, 1[$, we have shown

$$\begin{aligned} I_n(\alpha) &:= \int_0^1 \rho(v) f_n(\alpha, v) dv \\ &= \hat{I}_n(\alpha) + O(1/n) \\ &= \rho(\alpha) [1 + O(n^{-\kappa'/2})] + O(1/n) \\ &= \rho(\alpha) [1 + O(n^{-\kappa'/2})] \end{aligned}$$

and this ends the proof of the theorem for any $\alpha \in]0, 1[$. Note that on the interval $[\alpha_1, 1 - \alpha_1]$, with $\alpha_1 > 0$, the constant hidden in the O -term is uniform with respect to α .

Remark. It is probably possible to use the asymptotic normality of the α -quantile (as it is proven by Mosteller in [52]) for comparing more directly $\mathbf{QV}_\alpha(n)$ and $\mathbf{QQ}_\alpha(n)$. As we wish to obtain the precise remainder terms, we prefer to give the proof described here.

Now, we have proven all the assertions of Theorem 7.2. We now focus on the explicit expressions of the constants and prove Theorem 7.3.

7.6 Study of the asymptotic slope $\rho(\alpha)$.

7.6.1 An alternative expression for the asymptotic slope $\alpha \mapsto \rho(\alpha)$ of QuickVal.

Let $\alpha \in [0, 1]$. The asymptotic slope $\kappa(\alpha)$ for the mean number of key-comparisons, and the asymptotic slope $\rho(\alpha)$ for the mean number of symbol-comparisons are given by

$$\kappa(\alpha) = 2 \int_{\mathcal{T}} \phi_\alpha(u, t)^{-1} du dt, \quad \rho(\alpha) = 2\mathcal{J}[\phi_\alpha^{-1}] = \int_{\mathcal{T}} [\gamma(u, t) + 1] \phi_\alpha(u, t)^{-1} du dt, \quad (7.22)$$

where ϕ_α is defined as $\phi_\alpha(u, t) := t \vee \alpha - u \wedge \alpha$ for a parameter $\alpha \in [0, 1]$.

We are interested in giving a “short” expression for the asymptotic slope $\rho(\alpha)$ and we first extend the definition of ϕ_α for any $\alpha \in \mathbb{R}$, with

$$\phi_\alpha(u, t) = t - \alpha \quad (\text{for } \alpha \leq 0), \quad \phi_\alpha(u, t) = \alpha - u \quad (\text{for } \alpha \geq 1).$$

This leads to an explicit expression for κ , now defined on the whole real line, which involves an extension H of the entropy function h , usually only defined on the interval $[0, 1]$. More precisely, we write $\kappa(y) = 2[1 + H(y)]$ where $H(y)$ is an extension of the entropic function h , defined as

$$H(y) := \begin{cases} y \log \left(1 - \frac{1}{y} \right) & \text{if } y < 0 \\ h(y) = -y \log y - (1-y) \log(1-y), & \text{if } y \in [0, 1] \\ (y-1) \log \left(1 - \frac{1}{y} \right) & \text{if } y > 1. \end{cases} \quad (7.23)$$

And, with such an extension of the function κ , we obtain the following “short” expression for the asymptotic slope.

Proposition 7.17. *For a source \mathcal{S} with fundamental intervals $[a_w, b_w]$, the asymptotic slope of the QuickVal algorithm satisfies, for any $\alpha \in [0, 1]$,*

$$\rho(\alpha) = \sum_{w \in \Sigma^*} (b_w - a_w) \kappa \left(\frac{\alpha - a_w}{b_w - a_w} \right).$$

Proof. Starting with the expression of $\rho(\alpha) := \varpi(\alpha, 1)$, namely

$$\frac{1}{2} \rho(\alpha) = \sum_{w \in \Sigma^*} \int_{\mathcal{T}_w} [(\alpha \vee t) - (\alpha \wedge u)]^{-1} du dt,$$

we use the simple change of variables which maps \mathcal{T}_w onto \mathcal{T} , of the form $t = a_w + p_w t'$, $u = a_w + p_w u'$, and the relations

$$\begin{aligned} (\alpha \vee t) &= a_w + [(\alpha - a_w) \vee (p_w t')] = a_w p_w + p_w \left[\left(\frac{\alpha - a_w}{p_w} \right) \vee t' \right], \\ (\alpha \wedge u) &= a_w + [(\alpha - a_w) \wedge (p_w u')] = a_w p_w + p_w \left[\left(\frac{\alpha - a_w}{p_w} \right) \wedge u' \right], \end{aligned}$$

entail the equality

$$\frac{1}{p_w} (\alpha \vee t) - (\alpha \wedge u) = \left[\left(\frac{\alpha - a_w}{p_w} \right) \vee t' \right] - \left[\left(\frac{\alpha - a_w}{p_w} \right) \wedge u' \right],$$

yielding the final result. \square

7.6.2 Maximality of the asymptotic slope of the QuickVal algorithm at $\alpha = 1/2$.

It is not always true that the asymptotic slope $\rho(\alpha)$ be maximal at $\rho = 1/2$, even for a symmetric source. A clear counter-example is shown for the simplest source, the binary source, in Figure 7.1. The following result shows that this is true for any unbiased memoryless source over an alphabet of *odd* size.

Proposition 7.18. *For an unbiased memoryless source over an alphabet of odd size, the constant $\rho(\alpha)$ is maximized by $\alpha = 1/2$.*

Proof. For an unbiased memoryless source over an alphabet of finite size r , we can express $\rho(\alpha)$ in terms of κ , for any $\alpha \in \mathbb{R}$, as

$$\rho(\alpha) = \sum_{\ell=0}^{\infty} r^{-\ell} \rho_{\ell}(\alpha) \quad \text{with} \quad \rho_{\ell}(\alpha) := \sum_{k=0}^{r^{\ell}-1} \kappa(r^{\ell} \alpha - k).$$

We will prove that each $\rho_{\ell}(\alpha)$ is maximum at $\alpha = 1/2$. When r is odd, we let $r^{\ell} - 1 = 2J$, and we write the index $k \in [0..2J]$ as $k = -j + J$ with $j \in [-J, J]$, together with $r^{\ell} \alpha - k = (2J + 1)\alpha - J + j$. Now, for integer $J \geq 0$ we define

$$f_J(x) := \sum_{j=-J}^J \kappa(j + x), \quad x \in \mathbb{R} \tag{7.24}$$

so that

$$\rho_\ell(\alpha) := f_J(x) \quad \text{with} \quad x := (2J+1)\alpha - J.$$

It is then sufficient to study f_J on the interval $[-J, J+1]$ and prove that f_J is maximal at $x = 1/2$. This will prove that ρ_ℓ is maximal at $\alpha = 1/2$.

Fix $J \geq 1$ and write f as shorthand for f_J . First observe that

$$\sup_{x \in \mathbb{R}} f(x) = \sup_{x \in [0,1]} \sup_{k \in \mathbb{Z}} f(k+x). \quad (7.25)$$

We will prove the following two facts:

- (i) For any $x \in [0, 1]$ the expression $f(j+x)$ is non-increasing in integer $j \geq 0$.
- (ii) The expression $f(x)$ is maximized over $x \in [0, 1]$ by $x = 1/2$.

With (i), and the symmetry of $f(x)$ with respect to $x = 1/2$, it follows, for any $x \in [0, 1]$, that $f(j+x)$ is unimodal in $j \in \mathbb{Z}$ with maximum at $j = 0$. The lemma then follows from (ii) and (7.25).

To establish (i), let $j \geq 0$ and observe

$$f(j+1+x) - f(j+x) = \kappa(j+J+1+x) - \kappa(j-J+x). \quad (7.26)$$

If $j-J+x \geq 1/2$, then the difference (7.26) is ≤ 0 because $\kappa(x)$ is non-increasing for $x \geq 1/2$. On the other hand, if $j-J+x < 1/2$, then the difference (7.26) is upper-bounded by

$$\kappa(j+J+1) - \kappa(j-J) = \kappa(-j-J) - \kappa(j-J) \leq 0.$$

To establish (ii), we begin by calculating $f(x)$ explicitly when $x \in [0, 1]$. First, using the first line of (7.23), we find

$$\frac{1}{2} \sum_{j=1}^J \kappa(j+x) = J + \sum_{j=1}^J (1-j-x)[\log(j+x) - \log(j-1+x)].$$

Using Abel's transform, and introducing the function $g(y) := (1-y)\log y$, we obtain

$$\frac{1}{2} \sum_{j=1}^J \kappa(j+x) = J + g(J+x) + x \log x + \sum_{j=1}^{J-1} \log(j+x).$$

Next, using the symmetry of $\kappa(x)$ about $x = 1/2$, we observe

$$\frac{1}{2} \sum_{k=-J}^{-1} \kappa(j+x) = \frac{1}{2} \sum_{j=1}^J \kappa(j+1-x) = J + g(J+1-x) + (1-x) \log(1-x) + \sum_{j=1}^{J-1} \log(j+1-x).$$

Summing these last two expressions together with the expression for $\kappa(x)$ in the second line of (7.23) we arrive at

$$\frac{1}{2} f(x) = (2J+1) + [g(J+x) + g(J+1-x)] + \sum_{j=1}^{J-1} \log[(j+x)(j+1-x)]. \quad (7.27)$$

Of the three terms here, the first is constant, the third is unimodal with maximum at $x = 1/2$ (since each term in the sum has this property), and the second is maximized at $x = 1/2$ because $g(y)$ is concave for $y \geq 1$. (In fact, it is concave for all $y > 0$.) Thus (ii) is established, as is the proposition. \square

Some experiments performed by Philippe Flajolet showed that this property does not hold for every memoryless source, even if it is symmetric and built on an alphabet of odd size. However, we conjecture the following:

Conjecture. *Consider a symmetric memoryless source over an ordered alphabet Σ of odd size denoted by $\{0, 1, \dots, r-1\}$ and assume that the middle symbol $(r-1)/2$ is the most probable, namely $p_{(r-1)/2} \geq p_i$ for all $i \in \Sigma$. Then the constant $\rho(\alpha)$ is maximized by $\alpha = 1/2$.*

Conclusion of the chapter. We have conducted the analysis of the mean of the number of symbol comparisons of **QuickSelect** and its variants **QuickMin**, **QuickMax** and **QuickRand**. The analysis of **QuickSelect** is a bit different since we have to make a detour by its dual, the **Quickval** algorithm and use as an auxiliary tool the Laplace method.

Chapter 8

Conclusion and perspectives

This thesis first shows the applicability of the method of analysis, proposed in the paper[12]. We describe a new point of view on basic algorithms and their analysis which can be partially automatized. Our dream is to revisit all algorithms that are described in a standard textbook with this point of view and perform their realistic analysis.

This thesis offers a first step toward a double realistic analysis for the cost in the number of comparisons of sorting and searching algorithms. First we take into account the internal structure of the keys in the input of the algorithms (because such keys might have complex internal structure). From that point of view, the unit measure cost is no longer the key comparison but the symbol comparison. That helps to unify the measure of performance between algorithms based on key comparisons and the ones based on symbol comparisons by considering the same unit cost, i.e. the symbol comparison. Thus the results in this thesis provide a fairer comparison between key-based algorithms such as classical ones **QuickSort**, **QuickSelect** and symbol-based algorithms such as **radix-sort** or algorithms based on digital trees. Second, most of the analyses in literature suppose a uniform distribution of the input. In this thesis, we model the input of the algorithm by a probabilistic source. The source is characterized by its fundamental probabilities $\{p_w, w \in \Sigma^*\}$ with p_w the probability of generating a word with prefix w . Each infinite word is parametrized by a real number from the interval $[0, 1]$. This general model of source encompasses other important sources such as memoryless sources, Markov chain sources and dynamical sources.

The algorithms analyzed in this thesis are **QuickSort**, **QuickSelect**, **QuickMin**, **InsSort**, **BubSort** and **SelMin**. Robust algorithms are the ones for which the asymptotic estimates of both the mean number of key comparisons $K(n)$ and symbol comparisons $S(n)$ are of the same order. This is the case of **InsSort**, **QuickMin**, **SelMin** and **QuickSelect**. The ratios $\frac{S(n)}{K(n)}$ involves coincidence functions (between two words) of various types, respectively uniform coincidence, logarithmic-coincidence and min-coincidence and what looks like a fractal variation for **QuickSelect**. For other algorithms, $\frac{S(n)}{K(n)}$ is equal to $\frac{1}{2h(S)} \log n$ (**BubSort** and **QuickSort**). We remark that the same ratio appears between the lower bound for the number of symbol comparisons (proved in this thesis to be asymptotic to $\frac{1}{2h(S)} n \log^2 n$) and the lower bound for the number of key comparisons. This intriguing ratio $\frac{1}{2h(S)} \log n$ is not yet totally explained.

When we mix our analyses with the point of view of Seidel, we obtain an alternative proof of our results for **QuickSort** and **InsSort**.

Open problems and perspectives

As a mid-term project we wish to revisit **Binary-search** and **Minimum tournament** algorithms.

Binary search. This is a divide and conquer algorithm for finding a key in an sorted array. **Minimum tournament** searches the minimum key by comparing each adjacent keys and returning the relative minimum recursively. We model the input by the set of points $v_0 = 0 < v_1 < \dots < v_{n-1} < v_n = 1$ of $[0, 1]$ and a real $x \in [0, 1] \setminus \{v_0, \dots, v_n\}$. The algorithm returns the unique interval $[v_i, v_{i+1}]$ containing x .

Algorithm 1: A continuous version of binary search.

Input: a partition $v_0 = 0 < v_1 < \dots < v_{n-1} < v_n = 1$ of $[0, 1]$

Result: Find $i \in [0..n-1]$ such that $x \in [v_i, v_{i+1}]$

$\ell \leftarrow 0;$

$r \leftarrow n;$

if $\ell + 1 = r$ **then**

 | **return** ℓ

end

$m \leftarrow \frac{\ell+r}{2};$

if $x < v_m$ **then**

 | **return** RankDicho(x, ℓ, m)

end

else

 | **return** RankDicho(x, m, r)

end

As always, we begin the analysis by computing the mean number of local comparisons between two keys. However, one of two keys is always $M(x)$. To simplify the problem, we suppose that both x and the points v_p forming the partition are uniformly distributed. There are difficulties: when n is not a power of two the partitioning tree is more difficult to described (there is no “uniformity” with respect to n), even when n is a power of 2 the

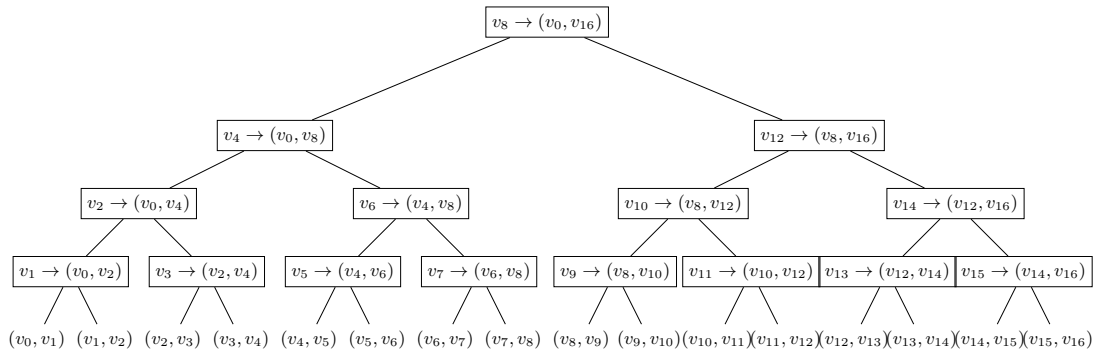


Figure 8.1: Example of the dichotomic partition tree for $n = 16$. The notation $v_p \rightarrow (v_\ell, v_r)$ means that in the partition process splits the interval $[v_\ell, v_r]$ in two subintervals at point v_p (called the pivot).

problem is intricate as expressions even for local comparisons $\pi_n(i, j)$ may be different for different n .

Minimum Tournament algorithm. This algorithm for selecting the minimum is modeled in the following way. We consider a set of n infinite words $X_i = M(u_i)$ (with $0 \leq i \leq n - 1$) uniformly and independently drawn from the same source. Again we have the problem that the shape of the tournament tree has leaves which may be on two different levels, and even when n is a power of 2, expressions are complicated.

These two algorithms are interesting since they do not enter strictly in our framework, and may give ideas or directions in order to extend it.

Appendix A

Generating functions

We define here the generating function which plays an important role in our analysis. For example, the Dirichlet series of the source $\Lambda(s)$ 2.1 is a generating function. The mean number of symbol comparisons of the algorithms in the Poisson model is also a generating function.

A.1 Generating function

According to the book [29], combinatorics is the study of finite structures built according to a finite set of rules. They are called discrete objects. Examples are graphs, words, trees, permutations, etc. One of the main problems in combinatorics is the enumeration of these objects and generating function is a central object in the symbolic enumeration methods.

A.1.1 Definitions

A combinatoric class is a finite or denumerable set of objects on which a size function is defined, satisfying the conditions: the size of an element is a non negative integer and the number of elements of any given size is finite. If \mathcal{A} is a class, one denotes by \mathcal{A}_n the set of objects in \mathcal{A} that have the size n . The cardinal of the set \mathcal{A}_n is denoted by A_n that is the number of objects in \mathcal{A} that have the size n . For example, we consider the set \mathcal{W} of words that are constructed from the alphabet $\{0, 1\}$:

$$\mathcal{W} = \{0, 1, 00, 01, 10, 11, \dots, 101001, \dots\}$$

The set of words of size n in \mathcal{W} is \mathcal{W}_n of cardinal W_n . We have

$$W_n = 2^n$$

because for each symbol of the word, there are two possibilities, 0 or 1. From these notations, we define the ordinary generating function related to a combinatoric class \mathcal{A} which is the formal power series:

$$A(z) = \sum_n A_n z^n.$$

The ordinary generating function (OGF) of a combinatoric class \mathcal{A} is the generating function of the numbers $A_n = \text{Card}(\mathcal{A}_n)$. Equivalently, the OGF of the class \mathcal{A} the combinatoric form

$$A(z) = \sum_{\alpha \in \mathcal{A}} z^{|\alpha|}.$$

This combinatoric form results from the fact that there are exactly A_n objects of size n in the class \mathcal{A} . In the previous example, the OGF of the class \mathcal{W} of binary words is

$$W(z) = \sum_{n=0}^{\infty} 2^n z^n = \frac{1}{1-2z}.$$

Some varieties of OGF is the exponential generating function (EGF) defined as

$$\hat{A}(z) = \sum_n A_n \frac{z^n}{n!},$$

and the Poisson generating function is defined by

$$\tilde{A}(z) = e^{-z} \sum_n A_n \frac{z^n}{n!} = e^{-z} \hat{A}(z).$$

As an example, we saw in Equation (5.17) of Chapter 5 that the exact mean number of symbol comparisons S_Z of sorting and searching algorithms has the form of a Poisson generating function:

$$\tilde{S}_Z = e^{-z} \sum_n S(n) \frac{z^n}{n!},$$

where S_n is the mean number of symbol comparisons over all permutations of size n .

A.1.2 Symbolic approach

The symbolic approach to combinatorial enumeration is a translation from general set-theoretic constructions to operations via generating functions. This admits a dictionary that includes a collection of core constructions: operation of union, cartesian product, sequence, set, multiset, and cycle. In the following, one denotes $\mathcal{A}, \mathcal{B}, \mathcal{C}$ three combinatorial classes.

Combinatorial sum The combinatorial sum \mathcal{A}_n of two classes \mathcal{B} and \mathcal{C} is the disjoint union of these two classes, i.e, the standard union of two classes that are supposed to be disjoint:

$$\mathcal{A}_n = \mathcal{B}_n \cup \mathcal{C}_n.$$

Because of the disjointness of two sets, one has

$$\text{Card}(\mathcal{A}_n) = \text{Card}(\mathcal{B}_n) + \text{Card}(\mathcal{C}_n), \quad \text{or, } A_n = B_n + C_n.$$

This gives the results:

$$A(z) = B(z) + C(z).$$

Cartesian product The construction $\mathcal{A} = \mathcal{B} \times \mathcal{C}$ forms all possible ordered pairs of elements of two classes \mathcal{B} and \mathcal{C} . From the combinatoric form of generating function, one has

$$A(z) = \sum_{\alpha \in \mathcal{A}} z^{|\alpha|} = \sum_{(\beta, \gamma) \in (\mathcal{B} \times \mathcal{C})} z^{|\beta|+|\gamma|} = \left(\sum_{\beta \in \mathcal{B}} z^{|\beta|} \right) \times \left(\sum_{\gamma \in \mathcal{C}} z^{|\gamma|} \right) = B(z) \cdot C(z).$$

Sequence construction If \mathcal{B} is a class, the sequence of \mathcal{B} , denoted by $SEQ(\mathcal{B})$ is defined as

$$SEQ(\mathcal{B}) = \{\epsilon\} + \mathcal{B} + \mathcal{B} \times \mathcal{B} + \mathcal{B} \times \mathcal{B} \times \mathcal{B} + \dots$$

with ϵ a neutral structure of size 0. The generating function of $SEQ(\mathcal{B})$ is deduced directly from the union and product relations above. Let $\mathcal{A} = SEQ(\mathcal{B})$, then:

$$A(z) = 1 + B(z) + B^2(z) + \dots = \frac{1}{1 - B(z)}.$$

where the geometric sum converges in the sense of formal power series since $[z^0]B(z) = 0$ assuming $B(z)$ does not contain elements of size 0.

Multiset construction Multisets are like finite sets but arbitrary repetitions of elements are allowed. The generating function notion of a multiset $\mathcal{A} = MSET(\mathcal{B})$ when \mathcal{A} is obtained by forming all finite multisets of elements from \mathcal{B} . We can define $MSET(\mathcal{B})$ as a quotient:

$$MSET(\mathcal{B}) := SEQ(\mathcal{B}) / \mathbf{R},$$

with \mathbf{R} , the equivalent relation of sequences:

$$(\alpha_1, \alpha_2, \dots, \alpha_r) \mathbf{R} (\beta_1, \beta_2, \dots, \beta_r)$$

iff there exists some arbitrary permutation σ of $[1, 2, \dots, r]$ such that for all j , $\beta_j = \alpha_{\sigma(j)}$.

Powerset construction The powerset class of a class \mathcal{B} , denoted by

$$\mathcal{A} = PSET(\mathcal{B}).$$

is defined as the class consisting of all finite subsets of class \mathcal{B} . The powerset class is equivalently the multisets that involve no repetitions.

Cycle construction The equivalence relation \mathbf{S} between sequences defined by

$$(\beta_1, \beta_2, \dots, \beta_r) \mathbf{S} (\beta'_1, \beta'_2, \dots, \beta'_r)$$

iff there exists a circular shift τ of $[1, \dots, r]$ such that for all j , $\beta'_j = \beta_{\tau(j)}$.

Cycles are sequences taken up to a circular shift of their components, the notation being $CYC(\mathcal{B})$:

$$CYC(\mathcal{B}) := ((SEQ(\mathcal{B}) \setminus \{\epsilon\}) / \mathbf{S}.$$

Sequences are grouped into equivalence classes according to the relation \mathbf{S} . For example, there are 4 cycles formed from sequences of length 3 over two types of objects $\{0, 1\}$:

- 000
- 001, 010, 100
- 110, 101, 011
- 111

This dictionary of OGF is represented in Figure A.1.

Sum	$\mathcal{A} = \mathcal{B} + \mathcal{C}$	$A(z) = B(z) + C(z)$
Cartesian product	$\mathcal{A} = \mathcal{B} \times \mathcal{C}$	$A(z) = B(z)\dot{C}(z)$
Sequence	$\mathcal{A} = \text{SEQ}(\mathcal{B})$	$A(z) = \frac{1}{1-B(z)}$
Powerset	$\mathcal{A} = \text{PSET}(\mathcal{B})$	$A(z) = \prod_{n \geq 1} (1 + z^n)^{B_n} = \exp \left(\sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{k} B(z^k) \right)$
Multiset	$\mathcal{A} = \text{MSET}(\mathcal{B})$	$A(z) = \prod_{n \geq 1} (1 - z^n)^{-B_n} = \exp \left(\sum_{k=1}^{\infty} \frac{1}{k} B(z^k) \right)$
Cycle	$\mathcal{A} = \text{CYC}(\mathcal{B})$	$A(z) = \sum_{k=1}^{\infty} \frac{\phi(k)}{k} \log \frac{1}{1 - B(z^k)}$

Figure A.1: Dictionary of OGF

Appendix B

Complex analysis

We will define basic notions in complex analysis such as complex number, analytic and meromorphic function, singularities and residue theorem (see also [68]).

B.1 Holomorphic function and Cauchy theorem

A complex is a number that has the form $z = x + iy$ where x is the real part, y the imaginary part of z : $x, y \in \mathbb{R}$ and i is the imaginary number that satisfies $i^2 = -1$. The set of complex numbers is noted \mathbb{C} and we note:

$$x = \Re(z), \quad y = \Im(z).$$

For $z \neq 0$, z admit a polar form $z = re^{i\theta}$ where r is the radius and θ the argument of z . We have

$$r = |z| = \sqrt{x^2 + y^2}, \quad \theta = \arctan\left(\frac{y}{x}\right).$$

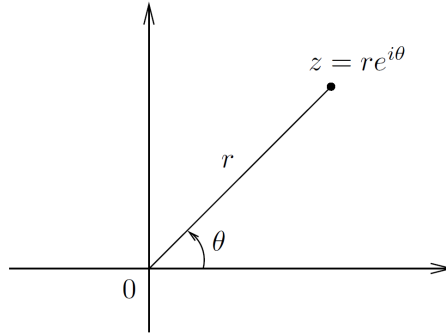


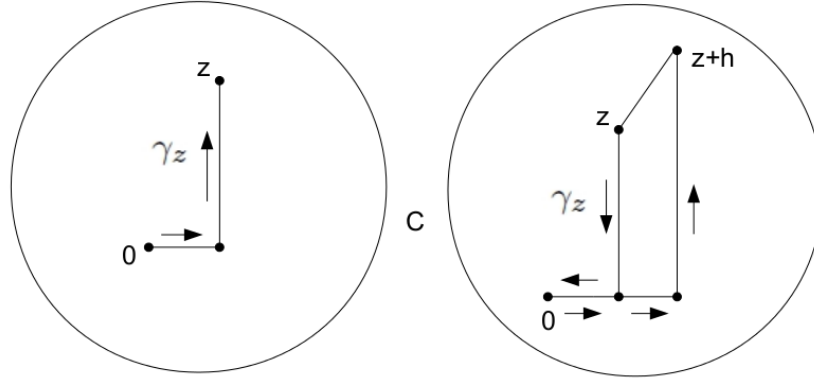
Figure B.1: The polar form of a complex number

Definition B.1 (Holomorphic function). *A function f is holomorphic at the point $z \in \mathbb{C}$ if the limit*

$$\lim_{h \rightarrow 0} \frac{f(z+h) - f(z)}{h}, \quad (h \in \mathbb{C})$$

exists. f is holomorphic in a open set if it is holomorphic at every point in the interior of this set.

Lemma B.2. *A holomorphic function in an open disc has a primitive in that disc.*

Figure B.2: The polygonal line γ_z

Proof. In the open disc \mathcal{C} where f is holomorphic we draw a polygonal line Γ_z (Figure B.2) that joins 0 to z by moving horizontally from 0 to $\Re(z)$ (the real part of z), then vertically from $\Re(z)$ to z . We pose

$$F(z) = \int_{\Gamma_z} f(\omega) d\omega.$$

We will prove that $F(z)$ is the primitive of f , i.e

$$\lim_{h \rightarrow 0} \frac{F(z+h) - F(z)}{h} = f(z).$$

We have:

$$\frac{F(z+h) - F(z)}{h} = \frac{1}{h} \left(\int_{\gamma_{z+h}} f(\omega) d\omega - \int_{\gamma_z} f(\omega) d\omega \right) = \frac{1}{h} \int_{\eta_h} f(\omega) d\omega,$$

where η_h is the straight line that link z to $z+h$. As f is continuous at z ,

$$\exists \psi(\omega), \quad \lim_{\omega \rightarrow z} \psi(\omega) = 0, \quad \text{such that} \quad f(\omega) = f(z) + \psi(\omega).$$

$$\int_{\eta_h} f(\omega) d\omega = \int_{\eta_h} (f(z) + \psi(\omega)) d\omega = f(z)h + \int_{\eta_h} \psi(\omega) d\omega.$$

Divide terms in both side by h and calculate the limit when h goes to 0, we have

$$\lim_{h \rightarrow 0} \frac{F(z+h) - F(z)}{h} = f(z) + \lim_{h \rightarrow 0} \frac{1}{h} \int_{\eta_h} \psi(\omega) d\omega.$$

However, we have the following estimates:

$$\left| \int_{\eta_h} \psi(\omega) d\omega \right| < \sup_{\omega \in \eta_h} |\psi(\omega)| |h| \quad \text{and} \quad \sup_{\omega \rightarrow z} \psi(\omega) = 0.$$

We deduce that

$$\lim_{h \rightarrow 0} \frac{1}{h} \int_{\eta_h} \psi(\omega) d\omega = 0 \quad \text{and} \quad \lim_{h \rightarrow 0} \frac{F(z+h) - F(z)}{h} = f(z).$$

We conclude that if f is holomorphic in \mathcal{C} , f has a primitive in \mathcal{C} . □

Sometimes the terms regular or complex differential are used instead of holomorphic. Holomorphic functions have some general properties:

- Contour integration: If f is holomorphic in Ω , for any closed path $\gamma \in \Omega$,

$$\int_{\gamma} f(z) dz = 0.$$

- Regularity: If f is holomorphic, f is indefinitely differentiable.
- Analytic continuation: If f and g are holomorphic in Ω and $f = g$ in an arbitrarily disc in Ω , then $f = g$ everywhere in Ω .

The first property is exactly the Cauchy theorem.

Theorem B.3. (*Cauchy theorem*) If f is holomorphic in a open set Ω then for any closed curve γ in Ω

$$\int_{\Gamma} f(z) dz = 0.$$

Proof. We already proved that if $f(z)$ is holomorphic in an open set Ω , it has a primitive $F(z)$ in Ω (Lemma B.2). Thus, it is obvious that the integral of $f(z)$ in any closed curve $\gamma \in \Omega$ is null. \square

Theorem B.4. (*Cauchy integral formula*) Suppose f is holomorphic in an open set that contains the closure of a disc D . If C denotes the boundary circle of this disc with the positive orientation, then:

$$f(z) = \frac{1}{2i\pi} \int_C \frac{f(\zeta)}{z - \zeta} d\zeta,$$

for any point $z \in D$.

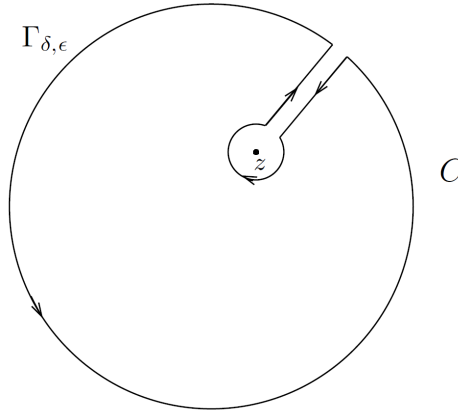


Figure B.3: Closed curve Γ and the keyhole $\Gamma_{\delta, \epsilon}$.

Proof Fix $z \in D$ and consider the keyhole $\Gamma_{\delta, \epsilon}$ (δ is the width of the corridor, ϵ is the radius of the small circle) that omits the point z .

Pose $F(\zeta) = \frac{f(\zeta)}{z - \zeta}$. As F is holomorphic away from z , $\int_{\Gamma_{\delta, \epsilon}} F(\zeta) d\zeta = 0$ by Cauchy's theorem. The integrals of $F(\zeta)$ over two sides of the corridor cancel out when $\delta \rightarrow 0$. The remaining part of the keyhole is the big circle C and the small circle C_ϵ centered at z and

of radius ϵ . Note C^+ the circle C in positive orientation and C_ϵ^- the circle C_ϵ in negative orientation.

$$F(\zeta) = \frac{f(\zeta) - f(z)}{z - \zeta} + \frac{f(z)}{z - \zeta}$$

As f is holomorphic in C , the first term on the right hand is bounded when $\epsilon \rightarrow 0$ and the integral over C_ϵ goes to 0 when $\epsilon \rightarrow 0$. We have:

$$\int_{\Gamma_{\zeta, \epsilon}} F(\zeta) d\zeta = \int_{C^+} F(\zeta) d\zeta + \int_{C_\epsilon^-} \frac{f(z)}{z - \zeta} d\zeta = 0$$

It is left to examine the integral over C_ϵ^- .

$$\int_{C_\epsilon^-} \frac{f(z)}{z - \zeta} d\zeta = f(z) \int_{C_\epsilon^-} \frac{d\zeta}{z - \zeta} = f(z) \int_0^{2\pi} \frac{-i\epsilon e^{-i\theta}}{\epsilon e^{-i\theta}} d\theta = -2i\pi f(z)$$

So it is clear that

$$\int_{C^+} \frac{f(\zeta)}{z - \zeta} d\zeta = 2i\pi f(z).$$

Definition B.5 (Analytic function). *A function $f(z)$ defined over a region Ω is analytic at a point $z_0 \in \Omega$ if, for z in some open disc centered at z_0 and contained in Ω , it is representable by a convergent power series expansion*

$$f(z) = \sum_{n=0}^{\infty} a_n (z - z_0)^n.$$

A function is analytic in a region Ω iff it is analytic at every point of Ω .

Furthermore, analyticity, holomorphic and complex differentiability are equivalent notions.

B.2 Meromorphic functions, poles and residue theorem

Given two analytic functions $f(z)$ and $g(z)$, the quotient $\frac{f(z)}{g(z)}$ ceases to be analytic at a point a where $g(a) = 0$. We now introduce meromorphic functions that are mild extensions of the concept of analyticity.

Definition B.6 (Meromorphic function). *A function $h(z)$ is meromorphic at z_0 iff, for z in a neighborhood of z_0 with $z \neq z_0$, it can be represented as $\frac{f(z)}{g(z)}$, with $f(z)$ and $g(z)$ being analytic at z_0 . In that case, it admits near z_0 an expansion of the form*

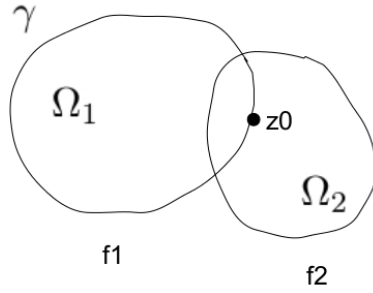
$$h(z) = \sum_{n \geq -M} h_n (z - z_0)^n.$$

If $h_{-M} \neq 0$ and $M \geq 1$, then $h(z)$ is said to have a pole of order M at $z = z_0$. The coefficient h_{-1} is called the residue of $h(z)$ at $z = z_0$ and is written as

$$\text{Res}[h(z); z = z_0].$$

A function is meromorphic in a region iff it is meromorphic at every point of the region.

Let $f_1(z)$ be analytical function defined over the interior region Ω_1 determined by a simple closed curve γ , and let z_0 be a point of the bounding curve γ . If there exists an analytic function $f_2(z)$ defined over some open set Ω_2 containing z_0 and such that $f_1(z) = f_2(z)$ in $\Omega_1 \cap \Omega_2$, one say that $f_1(z)$ is analytically continuable at z_0 and $f_1(z) = f_2(z)$ and f_2 is an immediate analytic continuation of f_1 .

Figure B.4: Analytic continuation of f_1 .

Definition B.7. Given a function defined in the region interior to the simple closed curve γ , a point z_0 on the boundary of (γ) of the region is a singular point or a singularity if f is not analytically continuable at z_0 .

The two functions $f(z) = \frac{1}{1-z}$ and $f(z) = \sqrt{1-z}$ are analytically continuable respectively in the regions $\Omega = \mathbb{C} \setminus 1$ and $(1, +\infty)$. They both have a singularity at $s = 1$ because

$$\lim_{z \rightarrow 1} f(z) = \infty.$$

and due to the branching character of the square root for g . Singularities often appear because the denominator of a function vanishes. A function $f(z)$ that is analytic at the origin, whose expansion at the origin has a finite radius of convergence R , necessarily has a singularity on the boundary of its disc of convergence, $|z| = R$. In particular, if the series expansion of $f(z)$ at the origin has non-negative coefficients and radius of convergence R , the point $z = R$ is a singularity of $f(z)$.

We define two types of singularities: removable singularity and poles.

Removable singularities A removable singularity of a function is a point singularity z_0 such that by defining $f(z_0) = 0$, the resulting extension becomes holomorphic in a full neighborhood of z_0 . For example, the function f defined as $f(z) = z$ for $z \neq 0$ is not defined at 0. Then, $z_0 = 0$ is a point singularity of f . By setting $f(0) = 0$, the function becomes continuous and entire. In this case, this is a removable singularity.

Poles A pole of a function f is a point singularity z_0 such that $\frac{1}{f}$ is holomorphic in a neighborhood of z_0 , except at z_0 its self. By defining $\frac{1}{f(z_0)} = 0$, $f(z)$ becomes holomorphic in a full neighborhood of z_0 . Given the function $g(z) = \frac{1}{z}$, g goes to ∞ as z approaches 0. $z_0 = 0$ is not a removable singularity but a pole of g . If f has a pole at z_0 , there exists a unique integer n and a non vanishing holomorphic function h such that:

$$f(z) = (z - z_0)^{-n} h(z)$$

As z_0 is a pole of f , it is a zero of $\frac{1}{f}$. There exists a non vanishing holomorphic function g such that $\frac{1}{f(z)} = (z - z_0)^n g(z)$. The result follows with $h(z) = \frac{1}{g(z)}$. The order of the pole is n . If $n = 1$, this is a simple pole. The function $h(z)$ has a power series expansion:

$$h(z) = A_0 + A_1(z - z_0) + \dots$$

so that

$$f(z) = (z - z_0)^{-n} (A_0 + A_1(z - z_0) + \dots).$$

This results in the following proposition:

Proposition B.8. *If a function f has a pole of order n at z_0 , then f can be expressed as:*

$$f(z) = \frac{a_{-n}}{(z - z_0)^n} + \frac{a_{-n+1}}{(z - z_0)^{n-1}} + \frac{a_{-n+2}}{(z - z_0)^{n-2}} + \cdots + \frac{a_{-1}}{(z - z_0)} + G(z)$$

where $G(z)$ is a holomorphic function in a neighborhood of z_0 .

The sum

$$\frac{a_{-n}}{(z - z_0)^n} + \frac{a_{-n+1}}{(z - z_0)^{n-1}} + \frac{a_{-n+2}}{(z - z_0)^{n-2}} + \cdots + \frac{a_{-1}}{(z - z_0)}$$

is called the principal part of f at the pole z_0 . The coefficient a_{-1} is the residue of f at that pole. We write $\text{Res}_{z_0} f = a_{-1}$.

We remark that all terms of order strictly greater than 1 in the principal part have a primitive in a deleted neighborhood of z_0 . Therefore if $P(z)$ denoted the sum above and C any circle centered at z_0 :

$$\frac{1}{2i\pi} \int_C P(z) dz = a_{-1} = \text{Res}_{z_0} f$$

Proposition B.9. *If f has a pole of order n at z_0 then*

$$\text{Res}_{z_0} f = \lim_{z \rightarrow z_0} \frac{1}{(n-1)!} \left(\frac{d}{dz} \right)^{n-1} (z - z_0)^n f(z)$$

This is an immediate consequence of the formula above of $f(z)$:

$$(z - z_0)^n f(z) = a_{-n} + a_{-n+1}(z - z_0) + \cdots + a_{-1}(z - z_0)^{n-1} + G(z)(z - z_0)^n.$$

Theorem B.10 (The residue formula). *Suppose that f is holomorphic in a open set containing a circle C and its interior, except for a pole at z_0 inside C . Then*

$$\int_C f(z) dz = 2i\pi \text{Res}_{z_0} f$$

Proof We can choose a keyhole contour that avoids the pole: $\int_C f(z) dz = \int_{C_\epsilon} f(z) dz$. The Cauchy's integral formula applies to a constant function shows that

$$a_{-1} = \frac{1}{2i\pi} \int_{C_\epsilon} \frac{a_{-1}}{z - z_0} dz.$$

Similarly, by using the corresponding formula for the derivatives, $\int_{C_\epsilon} \frac{a_{-1}}{(z - z_0)^k} dz = 0$, $\forall k > 1$.

This implies the desired result. The following proposition is a generalization of the residue formula when f has more than one pole inside the open set.

Proposition B.11. *Suppose that f is holomorphic in an open set containing a toy contour γ and its interior, except for poles at the points z_1, \dots, z_N inside γ . Then*

$$\int_\gamma f(z) dz = 2\pi i \sum_{k=1}^N \text{Res}_{z_k} f.$$

Appendix C

Upper and lower bounds

Analysis of algorithms is a large part of computational complexity theory. In theoretical analysis of algorithms, it is common to determine the asymptotic estimates of the complexity of algorithms. Therefore, the notations of Big O, Little o, Big Theta (Θ), Big Gamma (Ω) are used to the end. O , Ω , Θ -notations express respectively the lower bound, upper bound, both the lower and upper bounds since sometimes these two bounds are coincided.

- $g(x) = O(f(x)), x \rightarrow \infty$ if there exists a constant $A > 0$ and a real x_0 such that for $\forall x > x_0, |g(x)| < A|f(x)|$.
- $g(x) = o(f(x)), x \rightarrow \infty$ if for any constant $A > 0$, exists a real x_0 such that for $\forall x > x_0, |g(x)| < A|f(x)|$.
- $g(x) = \Omega(f(x))$ if there exists a constant $A > 0$ and a real x_0 such that for $\forall x > x_0, g(x) > Af(x)$.
- $g(x) = \Theta(f(x))$ if there exists two constants $A > 0, B > 0$ and a real x_0 such that for $\forall x > x_0, Af(x) < g(x) < Bf(x)$.

From the definition above, we define

- $O(f(n))$ the set of functions $g(n)$ such that $|\frac{g(n)}{f(n)}|$ is bounded from above as $n \rightarrow \infty$.
- $\Omega(f(n))$, the set of functions $g(n)$ such that $|\frac{g(n)}{f(n)}|$ is bounded from below as $n \rightarrow \infty$.
- $\Theta(f(n))$, the set of functions $g(n)$ such that $|\frac{g(n)}{f(n)}|$ is bounded from both below and above as $n \rightarrow \infty$.

For complexity studies, these notations allow implementation details to be hidden by ignoring constant factors. Since constant factors are ignored, the results are expressed in a simpler form than if more precise answer is sought. In the analysis of algorithms, O -notation and o -notation are used more widely than the others to express the asymptotic approximation, particularly to express small relative error terms. For example, the binary logarithm $\lg n$ and natural logarithm $\log n$ which are related by a constant can be both expressed as $O(\log n)$ in a complexity analysis.

Bibliography

- [1] AHO, A. V., HOPCROFT, J. E., AND ULLMAN, J. D. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974. (Cited on page 16.)
- [2] ANDERSON, D., AND BROWN, R. Combinatorial aspects of C.A.R Hoare’s Find algorithm. *Australasian Journal of Combinatorics* 5 (1992). (Cited on page 20.)
- [3] ASTRACHAN, O. L. Bubble sort: an archaeological algorithmic analysis. In *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education, 2003, Reno, Nevada, USA, February 19-23, 2003* (2003), pp. 1–5. (Cited on page 27.)
- [4] BALADI, V., AND VALLÉE, B. Distributional Analyses of Euclidean Algorithms. In *ALLENEX/ANALCO* (2004), pp. 170–184. (Cited on page 59.)
- [5] BALADI, V., AND VALLÉE, B. Euclidean algorithms are Gaussian. *Journal of Number Theory* 110 (2005), 331–386. (Cited on page 62.)
- [6] BENTLEY, J. L., AND MCILROY, M. D. Engineering a Sort Function. *Softw., Pract. Exper.* 23, 11 (1993), 1249–1265. (Cited on page 16.)
- [7] BENTLEY, J. L., AND SEDGEWICK, R. Fast Algorithms for Sorting and Searching Strings. In *Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, 5-7 January 1997, New Orleans, Louisiana*. (1997), pp. 360–369. (Cited on pages 1 and 16.)
- [8] BOSE, R., AND NELSON, R. A sorting problem. In *Journal of the ACM (JACM)* (1962), pp. 282–296. (Cited on page 27.)
- [9] BRASSARD, G., AND BRATLEY, P. *Algorithmics - theory and practice*. Prentice Hall, 1988. (Cited on page 16.)
- [10] CESARATTO, E., AND VALLÉE, B. Distribution of the average external depth for tries in dynamical sources context. In *Proceedings of the conference “Logic Computability and Randomness”* (2007), pp. 33–34. (Cited on page 64.)
- [11] CESARATTO, E., AND VALLÉE, B. Gaussian distribution of trie depth for dynamical sources. *Combinatorics, Probability & Computing* 24, 01 (2014), 54–103. To appear. (Cited on pages 7, 64, and 116.)
- [12] CLÉMENT, J., FILL, J. A., FLAJOLET, P., AND VALLÉE, B. The Number of Symbol Comparisons in QuickSort and QuickSelect. In *ICALP (1)* (2009), pp. 750–763. (Cited on pages 1, 2, 3, 7, 37, 125, and 151.)
- [13] CLÉMENT, J., FLAJOLET, P., AND VALLÉE, B. Dynamical Sources in Information Theory: A General Analysis of Trie Structures. *Algorithmica* 29, 1 (2001), 307–369. (Cited on pages 1 and 116.)

- [14] CLÉMENT, J., NGUYEN THI, T., AND VALLÉE, B. A general framework for the realistic analysis of sorting and searching algorithms. Application to some popular algorithms. In *STACS (2013)*, pp. 598–609. (Cited on page 3.)
- [15] CLÉMENT, J., NGUYEN THI, T., AND VALLÉE, B. Realistic analysis of the Quickselect algorithm. *To be appear in ToCS (2014)*, 598–609. (Cited on pages 3 and 124.)
- [16] CLÉMENT, J., THI, T. H. N., AND VALLÉE, B. Towards a realistic analysis of some popular sorting algorithms. *Combinatorics, Probability & Computing* 24, 1 (2015), 104–144. (Cited on pages 3 and 136.)
- [17] CORMEN, T. H., STEIN, C., RIVEST, R. L., AND LEISERSON, C. E. *Introduction to Algorithms*, 2nd ed. McGraw-Hill Higher Education, 2001. (Cited on pages 1, 15, 16, and 25.)
- [18] D. MCCracken, H. W., AND LEE, T. *Programming Business Computers*. John Wiley, 1959. (Cited on page 27.)
- [19] DE LA BRIANDAIS, R. File Searching Using Variable Length Keys. *AFIPS Western JCC (2000)*, 295–98. (Cited on page 115.)
- [20] DOLGOPYAT, D. On decay of correlations in Anosov flows. *Ann. of Math.* 147, 2 (1998), 357–390. (Cited on pages 62, 63, and 64.)
- [21] DOLGOPYAT, D. Prevalence of rapid mixing in hyperbolic flows. *Ergod. Th. & Dynam. Sys.* 18 (1998), 1097–1114. (Cited on pages 63 and 64.)
- [22] DONGARRA, J., AND SULLIVAN, F. Guest Editors introduction: The top ten algorithms. *Computing in science engineering*, 2:1 (2000), 22–23. (Cited on page 16.)
- [23] FILL, J. A., AND NAKAMA, T. Analysis of the Expected Number of Bit Comparisons Required by Quickselect. *Algorithmica* 58, 3 (2010), 730–769. (Cited on page 126.)
- [24] FLAJOLET, P. The ubiquitous digital tree. In *STACS 2006, 23rd Annual Symposium on Theoretical Aspects of Computer Science, Marseille, France, February 23-25, 2006, Proceedings (2006)*, pp. 1–22. (Cited on page 115.)
- [25] FLAJOLET, P., GOURDON, X., AND DUMAS, P. Mellin Transforms and Asymptotics: Harmonic Sums. *Theoretical Comput. Sci.* 144, 1–2 (June 1995), 3–58. (Cited on pages 80 and 119.)
- [26] FLAJOLET, P., ROUX, M., AND VALLÉE, B. Digital Trees and Memoryless Sources: from Arithmetics to Analysis. *Proceedings of AofA'10, DMTCS, proc AM (2010)*, 231–258. (Cited on page 55.)
- [27] FLAJOLET, P., AND SEDGEWICK, R. Mellin Transforms and Asymptotics: Finite Differences and Rice's Integrals. *Theor. Comput. Sci.* 144, 1&2 (1995), 101–124. (Cited on page 74.)
- [28] FLAJOLET, P., AND SEDGEWICK, R. Mellin transforms and asymptotics: Finite differences and rice's integrals. *Theor. Comput. Sci.* 144, 1&2 (1995), 101–124. (Cited on page 82.)
- [29] FLAJOLET, P., AND SEDGEWICK, R. *Analytic Combinatorics*. Cambridge University Press, 2009. (Cited on pages 74, 80, and 155.)

- [30] FLORES, I. Analysis of internal computer sorting. *Journal of the ACM (JACM)* 8, 1 (1961). (Cited on page 27.)
- [31] FREDKIN, E. Trie memory. *Communication ACM*, 3 (1960). (Cited on page 115.)
- [32] FRIAS, L., AND ROURA, S. Multikey Quickselect. *Algorithmica* 69, 4 (2014), 958–973. (Cited on page 1.)
- [33] FRIEND, E. Sorting on electronic computer systems. *Journal of the ACM (JACM)*, 3 (1956). (Cited on page 27.)
- [34] FROBENIUS, G. Ueber Matrizen aus nicht negativen Elementen. *Königl. Preuss. Akad. Wiss.* (1912), 456–477. (Cited on page 40.)
- [35] GRABNER, P. J., AND PRODINGER, H. On a Constant Arising in the Analysis of Bit Comparisons in Quickselect. *Quaestiones Mathematicae* 31, 4 (2008), 303–306. (Cited on page 126.)
- [36] GRAHAM, R. L., KNUTH, D. E., AND PATASHNIK, O. *Concrete mathematics - a foundation for computer science (2. ed.)*. Addison-Wesley, 1994. (Cited on page 14.)
- [37] HENNEQUIN, P. Combinatorial Analysis of Quicksort Algorithm. *ITA* 23, 3 (1989), 317–333. (Cited on page 19.)
- [38] HOARE, C. A. R. Algorithm 63: partition. *Commun. ACM* 4, 7 (1961), 321. (Cited on page 16.)
- [39] HOARE, C. A. R. Algorithm 64: Quicksort. *Commun. ACM* 4, 7 (1961), 321. (Cited on page 16.)
- [40] HOARE, C. A. R. Algorithm 65: find. *Commun. ACM* 4, 7 (1961), 321–322. (Cited on page 20.)
- [41] HOARE, C. A. R. Quicksort. *Comput. J.* 5, 1 (1962), 10–15. (Cited on pages 1 and 16.)
- [42] HUN, K. *Analysis of depth of digital trees built on general sources*. PhD thesis, University of Caen, 2014. (Cited on page 7.)
- [43] IVERSON, K. *Programming language*. John Wiley, 1962. (Cited on page 27.)
- [44] JACQUET, P. Trie structure for Graph Sequences. *Proceedings of AofA’14, DMTCS proc* (2014), 181–192. (Cited on page 71.)
- [45] JACQUET, P., AND SZPANKOWSKI, W. Analytical de-Poissonization and its applications. *Theoretical Computer Science* 201, 1-2 (1998), 1–62. (Cited on page 71.)
- [46] JACQUET, P., AND SZPANKOWSKI, W. Entropy computations for discrete distributions: towards analytic information theory. *IEEE International Symposium on Information Theory* (1998). (Cited on pages 71 and 116.)
- [47] KIRSCHENHOFER, P., PRODINGER, H., AND MARTINEZ, C. Analysis of Hoare’s FIND algorithm with Median-of-three partition. *Random Struct. Algorithms* 10, 1-2 (1997), 143–156. (Cited on page 1.)

- [48] KNUTH, D. E. Mathematical Analysis of Algorithms. In *IFIP Congress (1)* (1971), pp. 19–27. (Cited on pages 1 and 20.)
- [49] KNUTH, D. E. *The Art of Computer Programming, Volume I: Fundamental Algorithms, 3rd Edition*. Addison-Wesley, 1997. (Cited on page 16.)
- [50] KNUTH, D. E. *The Art of Computer Programming, Volume III: Sorting and Searching, 2nd Edition*. Addison-Wesley, 1998. (Cited on pages 16 and 27.)
- [51] MARTÍNEZ, C., PANHOLZER, A., AND PRODINGER, H. The analysis of approximate QuickSelect and related problems. (Cited on page 1.)
- [52] MOSTELLER, F. On some useful inefficient statistics. *Ann. Math. Statist.* 17, 4 (1946), 377–408. (Cited on pages 129 and 146.)
- [53] NÖRLUND, N. E. Leçons sur les équations linéaires aux différences finies. In *Collection de monographies sur la théorie des fonctions*. Gauthier-Villars, Paris, 1929. (Cited on page 72.)
- [54] NÖRLUND, N. E. *Vorlesungen über Differenzenrechnung*. Chelsea Publishing Company, New York, 1954. (Cited on page 72.)
- [55] PERRON, O. Zur Theorie der Matrices. *Mathematische Annalen* 64 (2) 64, 2 (1907), 248–263. (Cited on page 40.)
- [56] ROUX, M. *Théorie de l'information, séries de Dirichlet et analyse d'algorithmes*. PhD thesis, Université de Caen - Basse Normandie, 2011. (Cited on page 63.)
- [57] ROUX, M., AND VALLÉE, B. Information theory: Sources, Dirichlet series, and realistic analyses of data structures. In *WORDS* (2011), pp. 199–214. (Cited on page 64.)
- [58] RUELLE, D. Dynamical Zeta function and transfer operators. *Notices of the AMS* (2002), 199–214. (Cited on page 46.)
- [59] SEDGEWICK, R. *Quicksort*. Outstanding Dissertations in the Computer Sciences. Garland Publishing, New York, 1975. (Cited on page 16.)
- [60] SEDGEWICK, R. Quicksort with Equal Keys. *SIAM J. Comput.* 6, 2 (1977), 240–268. (Cited on page 16.)
- [61] SEDGEWICK, R. Implementing Quicksort Programs. *Commun. ACM* 21, 10 (1978), 847–857. (Cited on page 16.)
- [62] SEDGEWICK, R. *Algorithms in C, Parts 1–4*. Addison-Wesley, Reading, Mass., 1998. 3rd ed. (Cited on pages 1 and 3.)
- [63] SEDGEWICK, R. *Algorithms in C++ - parts 1 - 4: fundamentals, data structures, sorting, searching (3. ed.)*. Addison-Wesley-Longman, 1999. (Cited on page 27.)
- [64] SEDGEWICK, R., AND FLAJOLET, P. *An introduction to the analysis of algorithms*. Addison-Wesley-Longman, 1996. (Cited on page 16.)
- [65] SEIDEL, R. Data-Specific Analysis of String Sorting. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA* (2010), pp. 1278–1286. (Cited on pages 4, 104, 111, 112, and 120.)

- [66] SINGLETON, R. C. Algorithm 347: an efficient algorithm for sorting with minimal storage [M1]. *Commun. ACM* 12, 3 (1969), 185–187. (Cited on page 20.)
- [67] STANAT, D., AND MCALLISTER, D. *Discrete Mathematics in Computer Science*. Prentice-Hall, 1977. (Cited on page 27.)
- [68] STEIN, E. M., AND SHAKARCHI, R. *Complex analysis*. Princeton Lectures in Analysis, II. Princeton University Press, Princeton, NJ, 2003. (Cited on page 159.)
- [69] VALLÉE, B. Rice or Poisson-Mellin? *In preparation*. (Cited on page 116.)
- [70] VALLÉE, B. Opérateurs de Ruelle-Mayer généralisés et analyse des algorithmes d’Euclide et de Gauss. *Acta Arithmetica* 2, 81 (1997), 101–144. (Cited on page 47.)
- [71] VALLÉE, B. Dynamical Sources in Information Theory: Fundamental Intervals and Word Prefixes. *Algorithmica* 29, 1/2 (2001), 262–306. (Cited on page 37.)
- [72] VALLÉE, B. Note du Mini-Cours. école 2010 des Jeunes Chercheurs du GDR IM (Chambéry). Tech. rep., 2010. (Cited on page 76.)
- [73] VAN EMDEN, M. H. Increasing the efficiency of quicksort. *Commun. ACM* 13, 9 (1970), 563–567. (Cited on page 16.)
- [74] WEGNER, L. M. Quicksort for Equal Keys. *IEEE Trans. Computers* 34, 4 (1985), 362–367. (Cited on page 16.)
- [75] WHITE, B. Lecture for CS 154 course, online version on <http://web.ics.purdue.edu/cs154/lectures/lecture010.htm>, 2004. (Cited on page 28.)

Abstract We revisit classical textbook sorting or selecting algorithms under a complexity model that fully takes into account the elementary comparisons between symbols composing the records to be processed. Our probabilistic models belong to a broad category of information sources that encompasses memoryless (i.e., independent-symbols) and Markov sources, as well as many unbounded-correlation sources. Under this perspective, commonly accepted assertions, such as “the complexity of Quicksort is $O(n \log n)$ ”, are to be challenged, and the relative merits of sorting and searching methods relying on different principles (e.g., radix-based versus comparison-based) can be precisely assessed. For instance we establish that, under our conditions, the average-case complexity of QuickSort is $O(n \log^2 n)$ (rather than $O(n \log n)$, classically), whereas that of QuickSelect remains $O(n)$. In fact we propose a framework which allows to revisiting three sorting algorithms (QuickSort, Insertion Sort, Bubble Sort) and two selection algorithms (QuickSelect and Minimum Selection). For each algorithm a precise asymptotic estimate for the dominant term of the mean number of symbol comparisons is given where the constants involve various notions of coincidence depending on the algorithm. Explicit expressions for the implied constants are provided by methods from analytic combinatorics. As an aside, in our setting, we are able to derive a lower bound for the average number of symbol comparisons for algorithms solving the sorting problem and using usual comparisons between strings.

Keys words: Analysis of algorithms, Information theory, analytic combinatorics, dynamical source, transfer operator, sorting and searching algorithms, probabilistic distribution, generating function.

Résumé: On revisite les algorithmes de tri et de recherche classiques en considérant que les entrées de l'algorithme sont des mots infinis et en prenant compte de comparaisons de symboles entre des mots. Nous travaillons sous des modèles probabilistes différents pour lesquels les symboles sont générés par une source générale qui comprend, par exemple, la source sans mémoire, la chaîne de Markov ou même des sources avec les corrélations non bornées. De ce point de vue, une assertion telle que “la complexité de QuickSort est $O(n \log n)$ ” n'est plus vérifiée et il n'est pas aisé de comparer entre eux des algorithmes reposant sur des principes différents de comparaison. Dans ce cadre, nous trouvons que la complexité en moyenne pour le nombre de comparaisons de symboles de QuickSort devient $O(n \log^2 n)$ alors que celle de QuickSelect reste en $O(n)$. Nous proposons une méthode générale qui permet de revisiter trois algorithmes de tri QuickSort, Tri Insertion et Tri Bulles et deux algorithmes de sélection QuickSelect et Sélection du Minimum. Pour chaque algorithme, nous calculons les asymptotiques du nombre moyen de comparaisons de symboles. Les constantes pour les termes dominants sont reliées à des notions différentes de coïncidence et dépendantes des algorithmes. Nous empruntons des méthodes de la combinatoire analytique. Dans notre cadre, nous obtenons de plus une borne inférieure pour le nombre moyen de comparaisons de symboles effectuées par des algorithmes de tri utilisant la comparaison usuelle entre des chaînes de caractères.

Mots clés: Analyse des algorithmes, Théorie de l'information, Combinatoire analytique, source dynamique, opérateur de transfert, Algorithmes de tri et de recherche, distribution probabiliste, fonction génératrice.

