



**HAL**  
open science

## Trop d'algorithmes tue l'algorithmes ?

David Monniaux

► **To cite this version:**

David Monniaux. Trop d'algorithmes tue l'algorithmes ?. École d'ingénieur. Vaulx en Velin, France. 2022. hal-04827975

**HAL Id: hal-04827975**

**<https://hal.science/hal-04827975v1>**

Submitted on 9 Dec 2024

**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.

# Trop d'algorithmes tue l'algorithmes ?

David Monniaux

October 26, 2022



# Contents

Définition classique

Tri

Algorithmique classique

Modélisation

Apprentissage

Conclusion



# Qu'est-ce qu'un algorithme ?

## Larousse

« Ensemble de règles opératoires dont l'application permet de résoudre un problème énoncé au moyen d'un nombre fini d'opérations. Un algorithme peut être traduit, grâce à un langage de programmation, en un programme exécutable par un ordinateur. »

## Points importants

- ▶ règles **opératoires**
- ▶ nombre fini d'opérations
- ▶ résoudre un **problème**
- ▶ « implémentable » en un programme exécutable

# Cormen et al., algorithmes

« Voici une définition informelle du terme algorithme : **procédure de calcul** bien définie qui prend en entrée une valeur, ou un ensemble de valeurs, et qui donne en sortie une valeur, ou un ensemble de valeurs. Un algorithme est donc une séquence d'étapes de calcul qui transforment l'entrée en sortie.

L'on peut aussi considérer un algorithme comme un outil permettant de résoudre **un problème de calcul bien spécifié**. L'énoncé du problème **spécifie**, en termes généraux, la relation désirée entre l'entrée et la sortie. L'algorithme décrit une procédure de calcul spécifique permettant d'obtenir cette relation entrée/sortie. »



# Knuth, The Art of Computer Programming

The modern meaning for algorithm is quite similar to that of recipe, process, method, technique, procedure, routine, rigmarole, except that the word “algorithm” connotes something just a little different. Besides merely being a finite set of rules that gives **a sequence of operations for solving a specific type of problem**, an algorithm has five important features:

1. **Finiteness**. An algorithm must always terminate after a finite number of steps. [...]
2. **Definiteness**. Each step of an algorithm must be **precisely defined**; the actions to be carried out must be **rigorously and unambiguously** specified for each case. [...]

# Knuth, The Art of Computer Programming

3. Input. An algorithm has zero or more **inputs**: quantities that are given to it initially before the algorithm begins, or dynamically as the algorithm runs. [...]
4. Output. An algorithm has one or more outputs: quantities that have a **specified relation to the inputs**. [...]
5. Effectiveness. An algorithm is also generally expected to be **effective**, in the sense that its operations must all be sufficiently basic that they can in principle be done exactly and in a finite length of time by **someone using pencil and paper**.

# Knuth, The Art of Computer Programming

Let us try to compare the concept of an algorithm with that of a **cookbook recipe**. A recipe presumably has the qualities of finiteness [...] but it notoriously **lacks definiteness**. There are frequent cases in which a cook's instructions are indefinite: "Add a dash of salt." A "dash" is defined to be "less than 1% teaspoon," and salt is perhaps well enough defined; but where should the salt be added — on top? on the side? Instructions like "toss lightly until mixture is crumbly" or "warm cognac in small saucepan" are quite adequate as explanations to a trained chef, but an algorithm must be specified to such a degree that **even a computer can follow the directions**.



# Contents

Définition classique

**Tri**

Algorithmique classique

Modélisation

Apprentissage

Conclusion

# Spécification d'un tri par comparaisons

Entrée : un **tableau** de données

Sortie : un tableau de données **trié** par ordre croissant

# Spécification d'un tri par comparaisons

Entrée : un **tableau** de données  
une relation d'**ordre** totale (dit dans quel ordre mettre deux éléments)

Sortie : un tableau de données **trié** par ordre croissant

# Spécification d'un tri par comparaisons

Entrée : un **tableau** de données  
une relation d'**ordre** totale (dit dans quel ordre mettre deux éléments)

Sortie : un tableau de données **trié** par ordre croissant et qui contient **les mêmes données** que l'entrée

# Spécification d'un tri par comparaisons

Entrée : un **tableau** de données  
une relation d'**ordre** totale (dit dans quel ordre mettre deux éléments)

Sortie : un tableau de données **trié** par ordre croissant et qui contient **les mêmes données** que l'entrée **5 3 1 3**  $\rightarrow$  **5 1 1 3**

# Spécification d'un tri par comparaisons

Entrée : un **tableau** de données  
une relation d'**ordre** totale (dit dans quel ordre mettre deux éléments)

Sortie : un tableau de données **trié** par ordre croissant et qui contient **les mêmes données** que l'entrée avec la même multiplicité

# Remarques

- ▶ données et la relation d'ordre fixées  $\rightarrow$  résultat uniquement défini
- ▶ choix « politique »
  - ▶ décision de trier
  - ▶ choix de la relation d'ordre (p.ex., coefficients des notes d'un concours)
  - ▶ mais pas le choix de l'algorithme

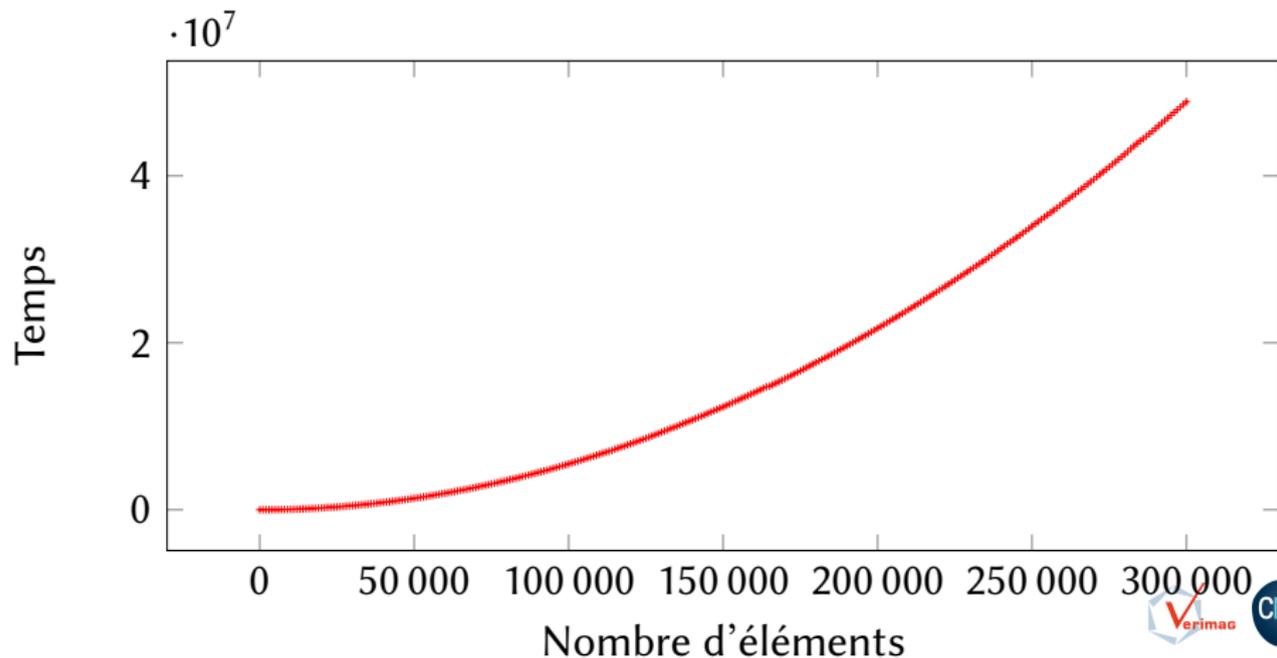
# Tri par sélection

## Tri par sélection

« On prend le plus petit et on l'échange avec l'élément en première position. On prend le plus petit dans le tableau restant et on l'échange avec l'élément en deuxième position. Et ainsi de suite. »

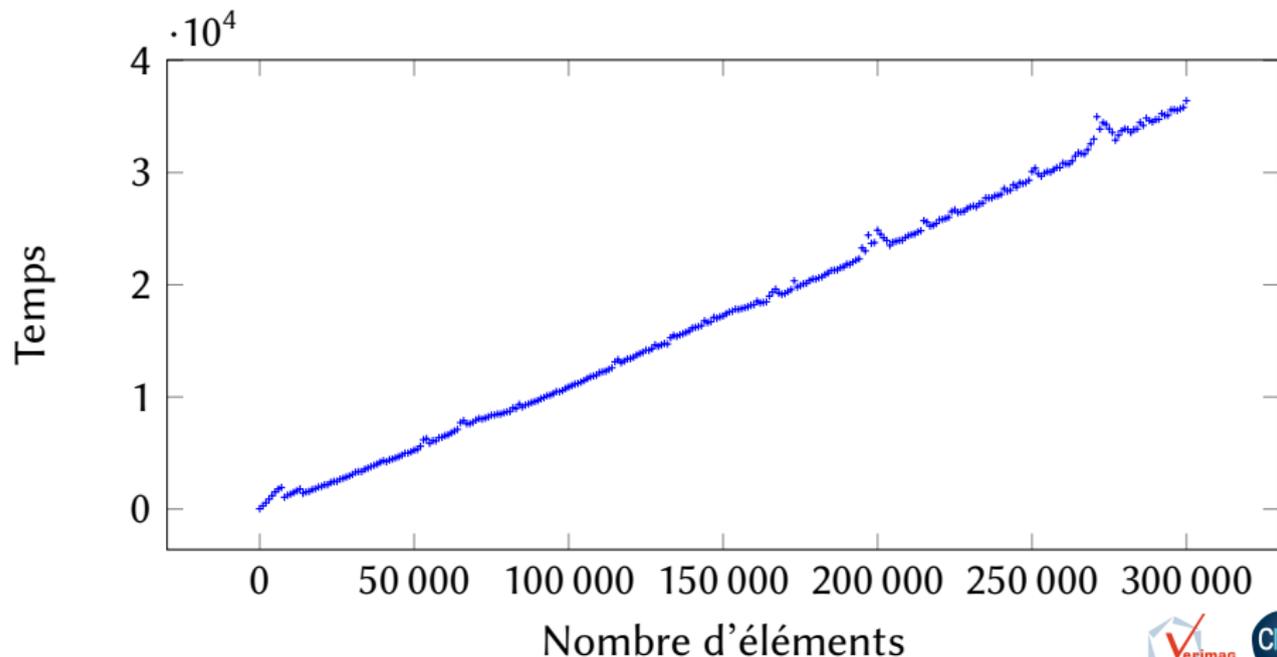
## Tri par sélection

« On prend le plus petit et on l'échange avec l'élément en première position. On prend le plus petit dans le tableau restant et on l'échange avec l'élément en deuxième position. Et ainsi de suite. »

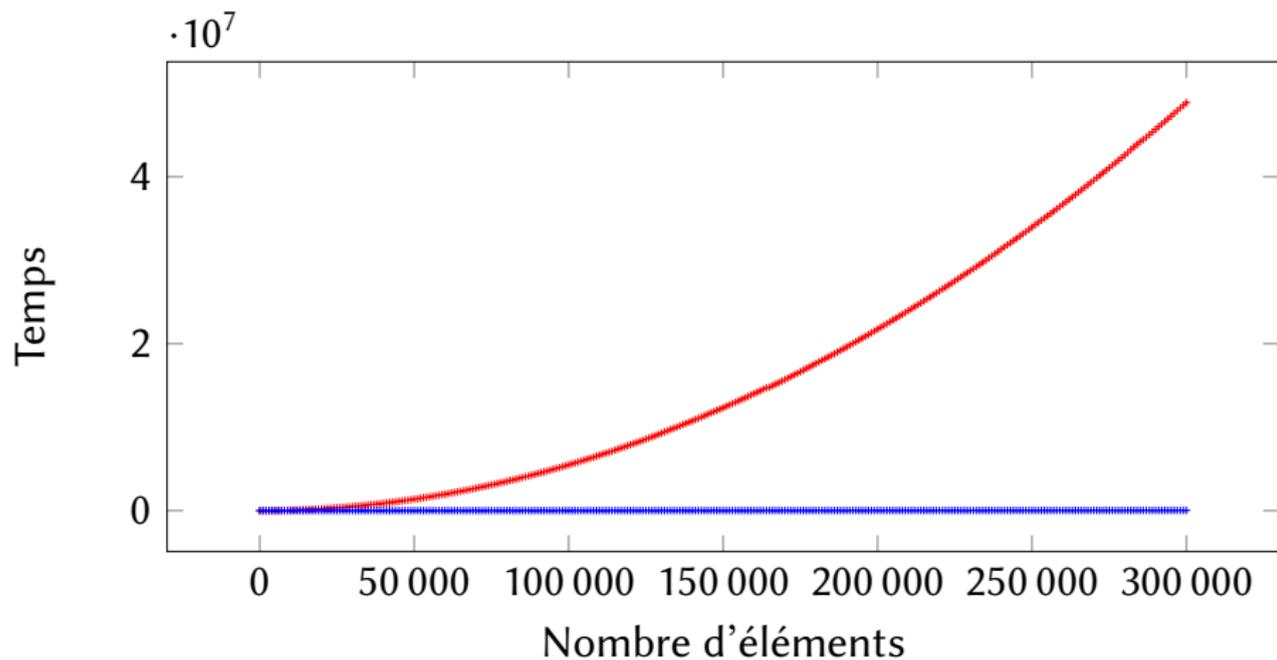


# Tri fusion

« On coupe le tableau en deux, on trie les deux moitiés et on fusionne les moitiés triées. (Si le tableau fait 0 ou 1 case on ne fait rien.) »



# Rôle de l'algorithmicien(ne)



# Contents

Définition classique

Tri

Algorithmique classique

Modélisation

Apprentissage

Conclusion

# Contenu des ouvrages d'algorithmique (Cormen, Knuth...)

- ▶ problèmes bien définis (définition mathématique précise)
- ▶ algorithme défini précisément
- ▶ **preuve mathématique** que l'algorithme répond au problème
- ▶ analyse du **coût du calcul** (temps, espace mémoire...)
- ▶ preuve mathématique de l'analyse
- ▶ parfois, données sur le « coût en pratique »

# Rôle classique de l'algorithmicien(ne)

Pour des problèmes **bien définis**,

- ▶ proposer des algorithmes qui **consomment moins** que l'état de l'art précédent
- ▶ démontrer qu'ils sont **corrects** (= donnent un bon résultat)
- ▶ étudier leur coût

# Autres rôles classiques de l'algorithmicien(ne)

- ▶ identifier des sous-problèmes plus simples à résoudre
- ▶ proposer des algorithmes **plus précis** (définition rigoureuse de l'erreur commise)
- ▶ proposer des algorithmes probabilistes et prouver leur probabilité de réussite
- ▶ étudier pourquoi il n'y a pas d'algorithmes « efficaces » pour traiter certains problèmes (théorie de la complexité algorithmique)

# Contents

Définition classique

Tri

Algorithmique classique

**Modélisation**

Apprentissage

Conclusion

# Les algorithmes répondent aux questions posées

## Question

« Ce programme est buggué, je veux qu'on le simplifie pour qu'on ne puisse plus atteindre la situation d'erreur. »

# Les algorithmes répondent aux questions posées

## Question

« Ce programme est buggué, je veux qu'on le simplifie pour qu'on ne puisse plus atteindre la situation d'erreur. »

## Réponse

« Remplacer le programme par un programme qui ne fait rien du tout. »

- ▶ c'est une simplification
- ▶ qui n'atteint pas la situation d'erreur
- ▶ on a oublié de spécifier qu'on voulait préserver la fonctionnalité du programme

# Remarques générales sur l'optimisation

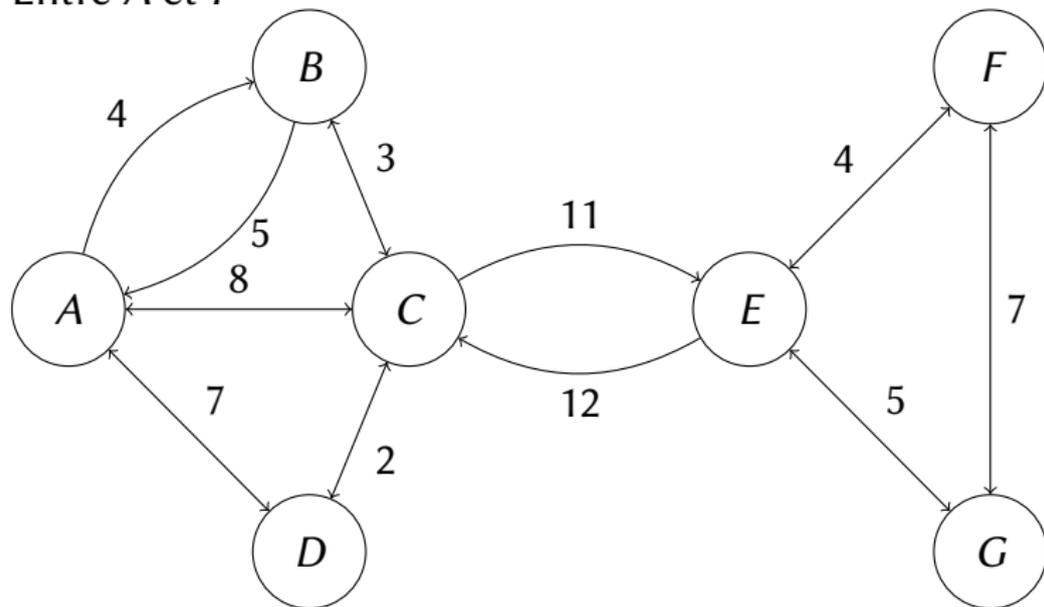
L'algorithme d'optimisation fournira une solution répondant aux contraintes si on peut les satisfaire.

En pratique

- ▶ contraintes oubliées, solutions « stupides »
- ▶ trop de contraintes rigides, pas de solution

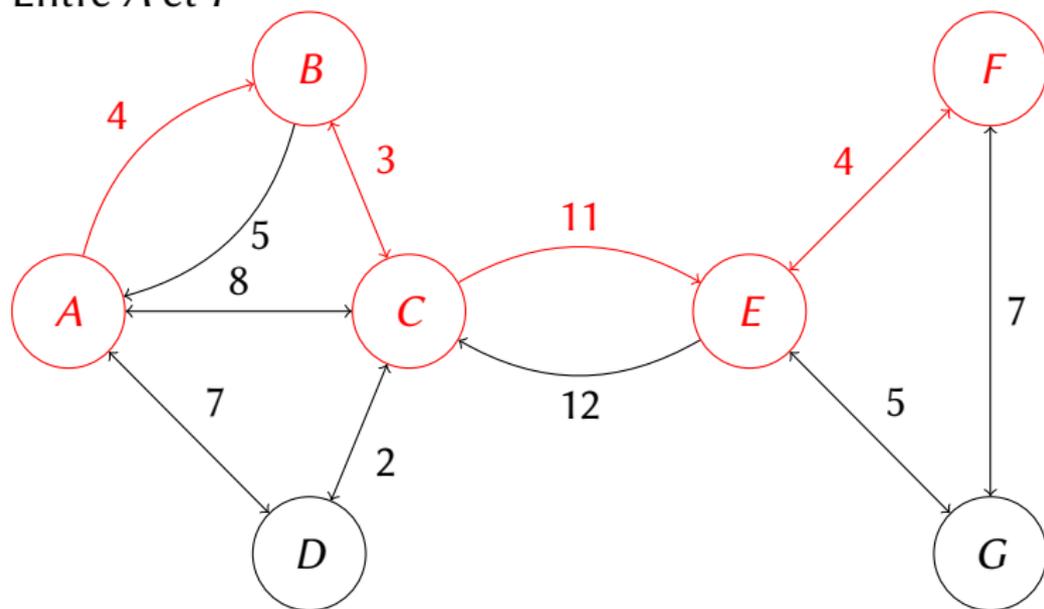
# Chemin de plus faible poids dans un graphe

Entre  $A$  et  $F$



# Chemin de plus faible poids dans un graphe

Entre *A* et *F*



# GPS

## Littérature

- ▶ Dijkstra
- ▶ A\*

GPS = variantes sur A\*

Graphe = points sur la carte, temps de trajet estimés sur segments de route

# Les erreurs de GPS

Pas dans l'algorithme de plus court chemin.

Dans le modèle = la carte avec les temps

# Contents

Définition classique

Tri

Algorithmique classique

Modélisation

**Apprentissage**

Conclusion

# Problèmes plus ou moins bien posés

## Problèmes bien définis

Donner le plus court chemin dans un graphe.

# Problèmes plus ou moins bien posés

## Problèmes bien définis

Donner **un** plus court chemin dans un graphe.

# Problèmes plus ou moins bien posés

## Problèmes bien définis

Donner plus court chemin dans un graphe.

## Problèmes mal définis

- ▶ donner le chemin qui prendra le moins de temps d'un point à un autre de la carte
- ▶ prédire quel prochain produit vous allez commander
- ▶ prédire si vous allez faire défaut sur votre crédit
- ▶ chercher des comportements précurseurs de terrorisme

# Solution: apprentissage

On renonce à formuler le modèle (voire la question).

On demande au système de les inférer via des **données**.

# Identification d'images

Apprentissage supervisé : on entraîne avec

- ▶ une collection d'images
- ▶ étiquetées avec ce qu'il y a dessus.

Avez-vous déjà travaillé à étiqueter ?

# Identification d'images

Apprentissage supervisé : on entraîne avec

- ▶ une collection d'images
- ▶ étiquetées avec ce qu'il y a dessus.

Avez-vous déjà travaillé à étiqueter ?

Exemple : demander d'identifier des bus, des bouches d'incendies, des feux tricolores...

# Exemple courant : reconnaissance de chiffres

On entraîne un réseau de neurones à reconnaître les chiffres imprimés ou manuscrits.

(Lecture de codes postaux, déclarations de revenus, etc.)

# « Algorithme raciste »

Dans la presse en 2018

La reconnaissance d'image de Google classe des humains à peau noire comme des gorilles.

# « Algorithme raciste »

Dans la presse en 2018

La reconnaissance d'image de Google classe des humains à peau noire comme des gorilles.

Entraînement sur une base de photos où les humains ont plutôt la peau claire ?

# « Algorithme raciste »

Dans la presse en 2018

La reconnaissance d'image de Google classe des humains à peau noire comme des gorilles.

Entraînement sur une base de photos où les humains ont plutôt la peau claire ?

Solution : interdire le label « gorille » même sur les photos de gorilles

# Les « algorithmes » issus de l'apprentissage

- ▶ problème à résoudre non défini mathématiquement
- ▶ modélisation en partie laissée à l'apprentissage
- ▶ difficultés à justifier les résultats (explicabilité = sujet de recherche)
- ▶ donner l'algorithme sans les données est peu pertinent

# Contents

Définition classique

Tri

Algorithmique classique

Modélisation

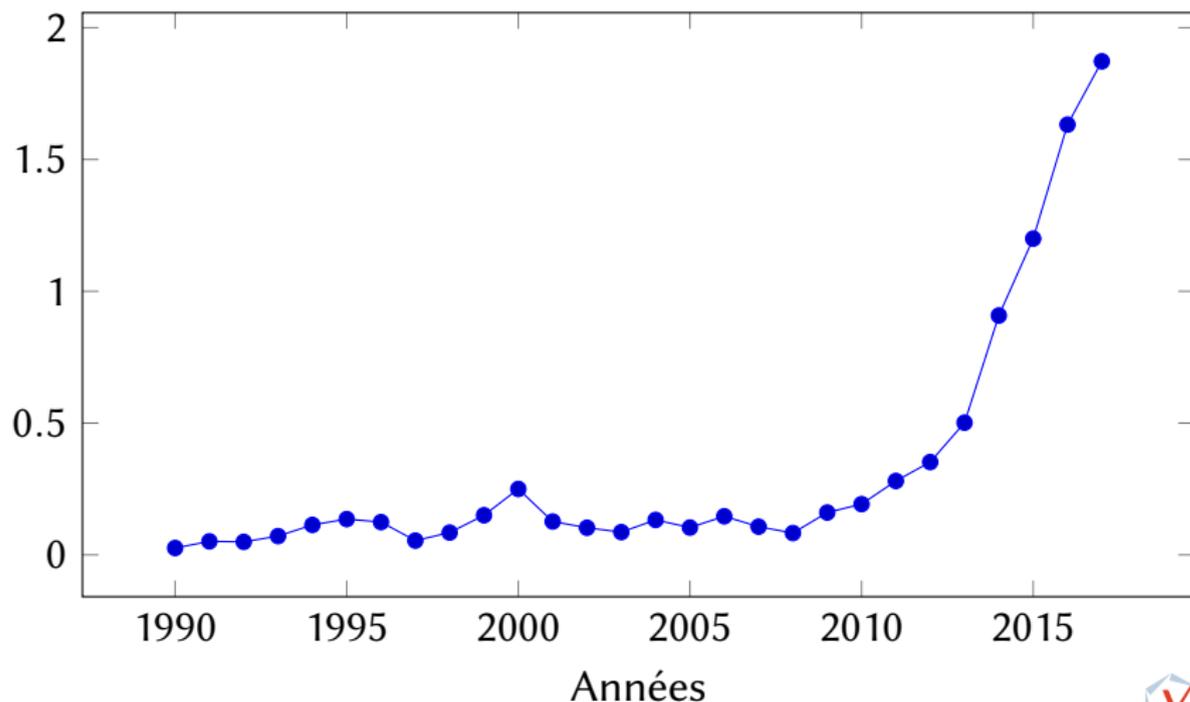
Apprentissage

Conclusion



# Un terme jadis confidentiel, maintenant répandu

Fréquence dans les grands quotidiens nationaux



# Gradation de problèmes

- ▶ problèmes bien posés mathématiquement
  - ▶ solution unique
  - ▶ plusieurs solutions possibles
- ▶ problèmes avec modélisation délicate
- ▶ problèmes où l'on apprend par l'exemple

# Le glissement de sens

## Sens classique

- ▶ résout un problème défini mathématiquement
- ▶ via un processus défini mathématiquement
- ▶ avec une preuve mathématique de cela

## Sens médiatique actuel

- ▶ logiciel, système informatisé complet
- ▶ boîte noire
- ▶ résultats arbitraires