



**HAL**  
open science

# Algorithmics in Arithmetic: Revealing algorithmic activities in a first-year arithmetic course for preservice teachers

Ulrich Kortenkamp

► **To cite this version:**

Ulrich Kortenkamp. Algorithmics in Arithmetic: Revealing algorithmic activities in a first-year arithmetic course for preservice teachers. Twelfth Congress of the European Society for Research in Mathematics Education (CERME12), Feb 2022, Bozen-Bolzano, Italy. hal-03748444

**HAL Id: hal-03748444**

**<https://hal.science/hal-03748444>**

Submitted on 9 Aug 2022

**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.

# Algorithmics in Arithmetic: Revealing algorithmic activities in a first-year arithmetic course for preservice teachers

Ulrich Kortenkamp

University of Potsdam, Germany; [ulrich.kortenkamp@uni-potsdam.de](mailto:ulrich.kortenkamp@uni-potsdam.de)

*The science of algorithms, that is, the design of algorithms and the analysis of their correctness, complexity or efficiency, is at the intersection of mathematics and computer science, as outlined in the scope and focus of the new thematic working group 11 of CERME12. However, one might question whether algorithmics is indeed a relevant topic for mathematics education. This paper investigates whether algorithmic activities are included in a first-year mathematics course for preservice elementary school teachers. Indeed, for 25 of 26 lectures in the course algorithmic activities could be identified.*

*Keywords: Arithmetic, preservice teacher education, elementary school mathematics, mathematics education.*

## Introduction

The new founded TWG11 that meets for the first time at CERME12 is focused on *algorithmics*, which is, according to the description of the TWG, the science of the design and analysis of algorithms. In this article, I try to give reasons for including such a group for a conference on mathematics education, by identifying algorithmic activities (to be defined later) in a lecture that is clearly addressing the needs of prospective elementary school mathematics teachers.

Before we can describe *algorithmic activities*, we have to explain what we mean by *algorithm*. In mathematics education, some prominent algorithms usually serve as defining examples, like the long division. More generally, according to Cormen (2009)<sup>1</sup>, “an **algorithm** is any well-defined computational procedure that takes some value, or set of values, as **input** and produces some value, or set of values, as **output**. An algorithm is thus a sequence of computational steps that transform the input into the output” (emphasis as in the original text). Of course, the output is not just any output, but should be the answer to the problem that is encoded by the input. Thus, the long division algorithm is an algorithm that receives a dividend and a divisor as input, and gives the quotient and remainder as output. An algorithm that produces the correct answer to any feasible input is called *correct* and is said to solve the corresponding computational problem (Cormen, 2009).

An algorithm needs to be created or *designed* by someone in order to be used. It has to be formulated or *specified precisely* in some language, for example in a natural language like English, in (pseudo code) as a description in or almost in a programming language, or in hardware that represents the algorithm (Cormen, 2009). If there is such a specification, it can be used to *carry out* the algorithm either manually or by a machine (usually a computer).

---

<sup>1</sup> While there is a vast amount of literature on algorithms in theoretical computer science, I refrain from citing all possible sources. For further sources that support the general statements on algorithms, please use the overview in the notes for Chapter 1 of Cormen (2009).

There might be several algorithms that solve the same computational problem. This calls for *comparing algorithms* for the same problem with respect to their efficiency in terms of *time* (number of computational steps needed to solve a problem) or *space* (usually the amount of memory needed to store intermediate results), or both. Here, it is important that these measures are dependent on the input, and some algorithms might be better in certain instances than others, even if they solve the same problem.

Other measures that might be surprising at first are *elegance* or *simplicity*. The latter can be defined by the number of different computational steps (or *elementary operations*) used by an algorithm, for example. *Elegance* is not as easy to define in general, as it is often the case for solutions to problems in mathematics that are called “elegant”. In Aigner & Ziegler (2010) this is discussed briefly for proofs of theorems that are worthy to be included in “The Book, in which God maintains the perfect proofs for mathematical theorems,” – and they just refuse to define or characterize what constitutes such a proof. At the same time, they claim for many of the proofs given, that they are “elegant”, again without a proper definition of elegant. For this paper, we can follow their lead and just conclude that it is possible to compare algorithms with respect to various measures.

In theoretical computer science there is much more work about algorithms that we cannot cover in this paper, but still we were able to identify several *algorithmic activities* in the last paragraphs:

- Design of algorithms: Creating an algorithm applicable to a class of problems
- Specification of algorithms: Describing the algorithm in a (formal) language
- Carrying out algorithms: Following the specification of an algorithm step by step
- Proving the correctness of algorithms: Finding arguments why the algorithm indeed solves the class of problems, either formally or pre-formal
- Comparing algorithms with respect to time (number of steps while carrying out), space (memory needed for bookkeeping during execution), elegance, or simplicity

In mathematics education, algorithms are often associated with mindlessly carrying out algorithms. Not only it is possible to *carry out algorithms* consciously, but this constitutes only a fraction of the possible algorithmic activities. For example, Krauthausen (1993) claims that it is a worthwhile activity to compare different approaches to written algorithms for multiplication, and Lisarelli et al. (2021) describe how 6<sup>th</sup> graders compare division algorithms.

There definitely is a common ground for algorithmics both in mathematics and computer science. In Knuth (1985) it is discussed to what extent mathematical thinking and algorithmic thinking coincide, without a final conclusion, but highlighting several occurrences of algorithmic thinking in a sample of mathematics books. Knuth chose nine math books and analyzed the content on page 100 of each book, which is a very intriguing methodological approach.

In our research we want to find out, which algorithmic activities are relevant to mathematics education. As a first approach to this question, we analyze a lecture that is part of the study program for prospective elementary school teachers. So, to refine the research question, we ask: *Which algorithmic activities can be identified in a first-year arithmetic course for prospective primary school teachers at a German university?*

## Context

The lecture that will be analyzed is part of the B.Ed. study program “Grundschulpädagogik” (elementary school pedagogy) at the University of Potsdam, Germany and has been created by the author and a colleague in 2010 at the University of Education Karlsruhe, Germany. It was held every year by author since then, and has been adapted to various settings in elementary teacher study programs at Karlsruhe (2010-2011), Potsdam (2014-2021) and the University of Halle-Wittenberg (2012-2013). The number of students in the course varied between 100 and more than 200, and it was always compulsory for the students. In Halle-Wittenberg, also prospective secondary school teachers were attending the lecture, and in Potsdam students of inclusive education both for elementary and secondary school have to attend the lecture. Starting 2021, the lecture has been made available as an Open Educational Resource (OER)<sup>2</sup>.

The universities in Halle-Wittenberg and Potsdam are the only universities involved in elementary teacher education in their respective states, so all elementary school teachers in the states Sachsen-Anhalt (Halle-Wittenberg) and later Brandenburg (Potsdam) who studied mathematics during these years should have been affected by the algorithmic activities contained in the course.

The course curriculum matches the regulations of the module in the accredited study programs. For accreditation of the programs it is necessary that they cover the topics and competences as given by the Standing Conference of the Ministers of Education and Cultural Affairs (Kultusministerkonferenz, 2019), which are based on the more detailed information of the German mathematical societies (DMV et al., 2008). While we only consider a single course as opposed to surveying all arithmetic courses in Germany, we still have reason to believe that the results are in line with other courses at other German universities, as the course adheres to the national standards, has been created by two lecturers with different biographies, was used in at least three universities and has been based on existing literature.

The course itself has been designed without an explicit focus on algorithmics. As such, it stands for a generic first-year mathematics course in teacher education, or any other elementary mathematics course in mathematics teacher education. Our goal is to identify algorithmic activities in existing courses, and not to design a course that includes algorithmic activities, thus proving the fact that algorithmic activities are relevant in mathematics teacher education.

## Data and Methodology

The data used for the study are the lecture slides that are provided as PDF files as accompanying material (1156 pages). For further details, also the original Keynote presentation (1446 slides) in the version of the academic year 2020/21 was available. Due to the Corona pandemic, the complete course has been recorded in short video clips of about 5-15 minutes that could be used asynchronously by the students. The complete video material (13+13 lectures for winter and summer semester, altogether 213 video clips with a total playing time of 28 hours, 19 minutes and 10 seconds,

---

<sup>2</sup> Currently available at <https://openup.uni-potsdam.de/course/view.php?id=65> through a guest login. A full version of the course in German will be published later.

was available for in-depth analysis in cases where the presentation slides did not show the activities in detail.

As a second data source, the accompanying exercises and the interactive materials used in demonstrations and for self-explorations by the students is available. However, this study focuses on the content that was presented through the video lectures.

The material was reviewed by the author and each part of the lecture was classified depending on whether any algorithmic activity as described above could be identified. As our research question is not of quantitative nature, we refrained from asking a second person to code the data. Also, we are aware of the limitation that the author of the course and this paper is the same person.

The data was coded on a per-slide basis, using the slides and videos for clarification in some cases. For each video it was noted whether algorithmic activities could be identified on the slides. If so, they were categorized in the categories *Design (D)*, *Formulation (F)*, *Carrying out (CAR)*, *Correctness proof (COR)*, and *Comparison or Analysis (COM)*. Both the activity and the page number in the PDF slides were noted. The involved algorithms were recorded as well, and in some cases additional remarks. After coding, the data was aggregated for each episode in both series.

## Results

The number of clips that contained algorithmic activities is shown in the table below, together with a short description of the algorithm involved in these activities.

**Table 1. Algorithmic Activities identified in the lecture on arithmetic (S=Series, E=Episode)**

S	E	German title of episode	Algorithms used	D	F	CAR	COR	COM
1	1	Mengen	Set Product		2		1	
1	2	Aussagen	Boolean operations / variables / loops	3	3	3	2	2
1	3	Beweisbedürfnisse und Beweistechniken	Find number not divisible	2	1	1	1	
1	4	Was machen wir gleich?	Change of representation / Completing a relation to an equivalence relation			2		
1	5	Everybody needs somebody to love - Abbildungen und Zahlen	Functions in General	1				
1	6	Zählen von Kardinalzahlen mit Ordinalzahlen	Counting and Peano	2	2	3	2	2
1	7	Vollständige Induktion	Induction / Zone theorem	1		1	2	1
1	8	Rechnen - Addition und Subtraktion	Addition by counting	2	3	1	2	1
1	9	Rechnen - Multiplikation und Division	Multiplication as repeated addition / division in various ways	4	4	2	4	2
1	10	GZSZ: Große Zahlen, Schöne Zahlen	Repeated bundling, base change	2	4	3	2	1
1	11	Rechnen in Stellenwertsystemen I: Addition und Subtraktion	Base change	1	4	4	2	2

S	E	German title of episode	Algorithms used	D	F	CAR	COR	COM
1	12	Mach mal Schräggitter & Wir bauen uns einen Computer	Written multiplication / hardware addition	4	4	3	2	3
1	13	Division in Stellenwertsystemen	Long division	3	3	5	2	3
2	1	Die ganzen Zahlen $\mathbb{Z}$	Multiplication of whole numbers (sign rules)				1	
2	2	Teilbarkeit, Teilen und Reste	Divisibility rules	3	6	2	4	4
2	3	Der Hauptsatz der Arithmetik	Euclidean algorithm etc.	4	3	3	4	2
2	4	Der Chinesische Restsatz	Chinese remainder theorem	3	2	1	4	1
2	5	Von $\mathbb{Z}$ nach $\mathbb{Q}$	Solving linear equations	1	1		1	1
2	6	Wie viele Rationale Zahlen gibt es?	Counting fractions	1	1	1		
2	7	Bruchrechnung in der Schule – Bruchzahlen	–					
2	8	Bruchrechnung in der Schule – Addition und Subtraktion	Addition	1	1	1		1
2	9	Bruchrechnung in der Schule – Multiplikation und Division & Dezimalzahlen	Division of fractions	1	1		1	
2	10	小数と分数 – Eine Reise durch die Klasse 3-6 in Japan	Multiplication algorithm	1	1		1	
2	11	Alles wird eins – Arithmetische und Geometrische Folgen	Representation of fractions	1	2	3	3	1
2	12	Reelle Zahlen	Approximation using Heron and nested intervals	2	4	2	3	2
2	13	$\mathbb{R}$ gebe mir ein $\mathbb{C}$	Algorithmic view on coordinates		1			1
TOTAL				<b>43</b>	<b>53</b>	<b>41</b>	<b>44</b>	<b>30</b>

The data shows, that indeed all but one lecture feature algorithmic activities. In S2E7 fractions and their representation in school are introduced and discussed. In that lecture, no operations with fractions are used, so it is difficult to describe an algorithm that is based on several steps using operations.

Most algorithms are designed prior to their formulation. Instead of coming out of the blue (or the textbook), algorithms are depicted as something that is created by someone. We see that it is hardly the case that algorithms are just used without a proper introduction, that is, a focus on their design. Also, most algorithms are discussed for their correctness, at least in part.

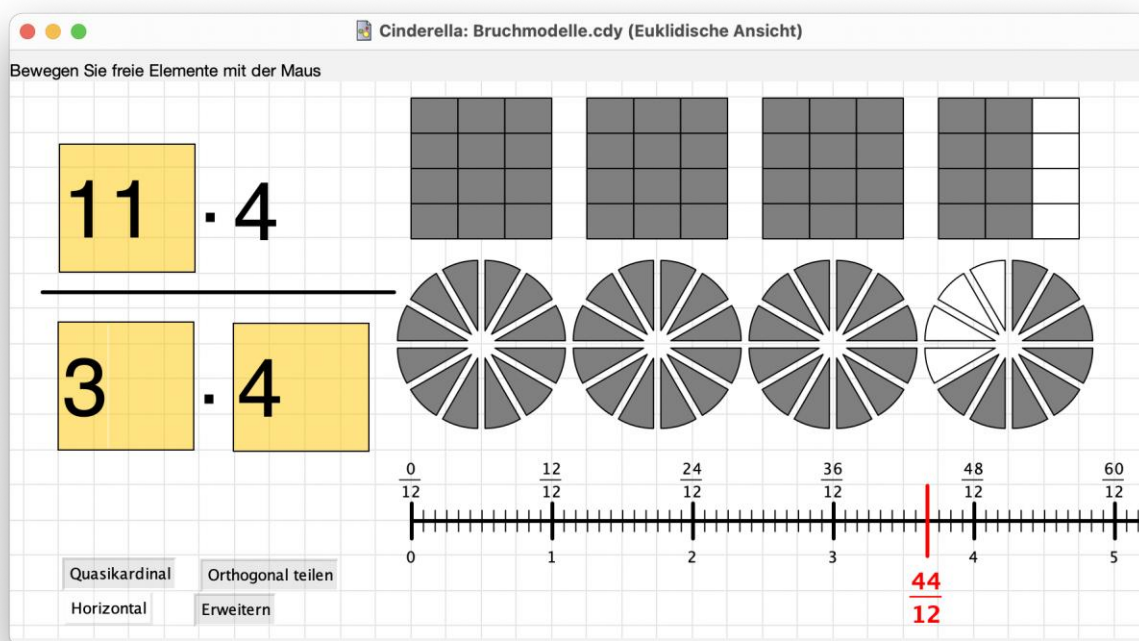
The comparison and analysis of algorithms is the activity that happens the least. This can be related to the fact that in many situations standard algorithms exists.

In S2E7–S2E10 less algorithmic activities can be seen. These three lectures focus on teaching fractions in school, both in Germany and Japan. They rely on typical content found in textbooks, and do not introduce new mathematical content, but mostly subject-specific pedagogical content. Still, algorithmic activities are highlighted whenever possible.

An activity that was found in the additional material is a homework project called “Maths around the world”. It takes place between the first and second semester of the lecture, and students are asked to find as many as possible interesting ways to do written calculation. Usually, they ask people from other generations (their grandparents) or friends from all over the world, or they have a migration background themselves and can report about their own experience in school. This activity is focused on analyzing and comparing algorithms, and shall compensate for the fact that most of the content in the lecture is being presented instead of experiencing it in a constructivist manner.

## Discussion

As can be seen from the results, algorithmic activities are indeed integrated into all lectures but one. This gives rise to the question whether S2E7, the lecture on fractions and their models, could be enhanced with algorithmic activities, too. One possible activity could be an algorithm that creates models from fractions (or vice versa). Further inspection shows that this would be in line with a digital activity that is available in the moodle course and has been used to create the lecture slides as well. Figure 1 shows a Cinderella (Richter-Gebert & Kortenkamp, 2012) based interactive construction that students can use to explore fractions. The construction of such a representation can be described through an algorithm, which can be discussed in the lecture.



**Figure 1. A dynamic representation of various fraction representations used in S2E7. The yellow fields can be changed using the mouse or keyboard and the representations change accordingly.**

Although there are lots of algorithmic activities that are already incorporated in the lecture, we ask whether it is possible to do more comparison and analysis. The comparison of algorithms is a very important element of the “Maths around the world” activity, where a lot of different algorithms for the same tasks are collected by the students. The question whether all algorithms lead to the same

result concerns the correctness, but students identify differences like a faster execution, a better understandability, or a more general or more specialized approach. This also leads to the important insight that they should appreciate and embrace approaches that their future students in the classroom already know from their parents or by their own invention. Another very prominent algorithm comparison takes place in S2E12, where square roots of numbers are approximated through Heron's algorithm and a traditional nested intervals approach. Students experience again that both algorithms lead to the same result (and thus are equivalent), but Heron's method finds an approximation with a given accuracy much faster. This introduces students to questions about the efficiency of algorithms that constitute an important part of computer science and shows that not only the problem itself can have a certain complexity, but it also depends on the algorithm that is used to solve it. So, it would be great to find more opportunities for comparison. Unfortunately, this introduces another problem, as the teacher students would have to learn more algorithms and might complain about additional content for their examination.

## **Conclusion and Future Work**

All in all, we can state that algorithmic activities are indeed an integral part of the arithmetic course we analyzed. While this could have been caused by background of the author of this paper, who also created the course, it still shows that algorithmic activities can be included in arithmetic organically.

The results from this study will be used to redesign the course, again. In particular, the homework assignments will be redesigned to include more algorithmic activities that are carried out by the students themselves, complementing the algorithmic activities that they experience during the lecture or in the video clips.

As a further step, we will apply the same methodology to other lectures in the B.Ed./M.Ed. program in mathematics education, to find out whether the results are specific to arithmetic – due to its computational roots – or can be generalized to other parts of mathematics as well. It is worthwhile to analyze other courses, in other subject areas like geometry, stochastics or algebra, from other authors and to see whether they already contain algorithmic activities, and to see how the inclusion could be achieved or strengthened based on the examples found in this course.

## **Acknowledgment**

The “Arithmetic and its didactics” course would not have been possible without the discussions with and contributions of the teaching and student assistants who accompanied me over several years of refinement of the original course created by Mutfried Hartmann, University of Education Karlsruhe and myself. In particular, I want to mention Johanna Goral and Karen Reitz-Koncebovski, who analyzed the content of the course and forced me to disclose the implicit design principles of the course, and Felix Grohmann, who helped to create the OER version of course.

## **References**

Aigner, M., & Ziegler, G. M. (2010). *Proofs from THE BOOK*. Springer. <https://doi.org/10.1007/978-3-642-00856-6>

Cormen, T. H. (Ed.). (2009). *Introduction to algorithms* (3rd ed). MIT Press.



- DMV, GDM, & MNU. (2008). Standards für die Lehrerbildung im Fach Mathematik. Empfehlungen von DMV, GDM und MNU. *Mitteilungen der Deutschen Mathematiker-Vereinigung*, 16(3), 149–159. <https://doi.org/10.1515/dmvm-2008-0057>
- Knuth, D. E. (1985). Algorithmic Thinking and Mathematical Thinking. *The American Mathematical Monthly*, Vol. 92, Nr. 3, 170–181.
- Krauthausen, G. (1993). Kopfrechnen, halbschriftliches Rechnen, schriftliche Normalverfahren, Taschenrechner: Für eine Neubestimmung des Stellenwertes der vier Rechenmethoden. *Journal für Mathematik-Didaktik*, 14(3–4), 189–219. <https://doi.org/10.1007/BF03338792>
- Kultusministerkonferenz. (2019). *Ländergemeinsame inhaltliche Anforderungen für die Fachwissenschaften und Fachdidaktiken in der Lehrerbildung. Beschluss der Kultusministerkonferenz vom 16.10.2008 i. D. F. vom 16.05.2019.* [https://www.kmk.org/fileadmin/veroeffentlichungen\\_beschluesse/2008/2008\\_10\\_16-Fachprofile-Lehrerbildung.pdf](https://www.kmk.org/fileadmin/veroeffentlichungen_beschluesse/2008/2008_10_16-Fachprofile-Lehrerbildung.pdf)
- Lisarelli, G., Baccaglioni-Frank, A., & Di Martino, P. (2021). From how to why: A quest for the common mathematical meanings behind two different division algorithms. *The Journal of Mathematical Behavior*, 63, 100897. <https://doi.org/10.1016/j.jmathb.2021.100897>
- Richter-Gebert, J., & Kortenkamp, U. H. (2012). *The Cinderella.2 Manual*. Springer. <https://doi.org/10.1007/978-3-540-34926-6>