ALGORITHMS BEFORE COMPUTERS

Michel ROELENS

Faculty of Teacher Education, University Colleges Leuven-Limburg (University of Applied Sciences), Diepenbeek, Belgium

Maria-Boodschaplyceum (secondary school), Brussels, Belgium

Uitwiskeling (magazine for mathematics teachers), Belgium

Michel.Roelens@ucll.be

ABSTRACT

In the new curricula that are now gradually being implemented in secondary schools in Flanders, the Dutch-speaking part of Belgium, there is a shift towards mathematics as a part of STEM (Science, Technology, Engineering, Mathematics) and towards computational thinking in mathematics: programming, logic and electronic gates, graph theory, and for some pupils linear programming. Meanwhile, the new curricula contain less geometry and fewer references to history and art. As a first reaction, I welcome the introduction of graph theory, but I regret the loss of some beautiful parts of geometry. Instead of complaining, I try to relate the new subjects with geometry and the history of mathematics. There have always been algorithms in mathematics, long before there were electronic computers that could be programmed. Moreover, geometrical and visual thinking can be used to discover many algorithms.

In this article, I would like to illustrate historical and geometrical aspects of algorithms with some examples: the calculation by hand of the digits of a square root and the algorithm for constructing a Eulerian circuit in a multigraph in which the degree of each vertex is even.

1 Introduction

1.1 My personal situation

I teach mathematics - and especially geometry - on a half-time basis to future secondary school teachers. In our teacher training in Flanders, prospective teachers for pupils aged 12 to 16 years take two subjects: my students will become teachers of for example mathematics and biology, or mathematics and French. I also teach mathematics to pupils aged 16 to 18 years, in a secondary school in the centre of Brussels, three days a week. At night and weekends, since 1984, I work on the magazine *Uitwiskeling* for mathematics teachers.

1.2 Recent tendencies in Flemish curricula

New curricula are introduced in secondary education in Flanders. Due to a decision by the Constitutional Court, these curricula are likely to be reduced, but I will not go into these political issues here.

The new curricula present the following trends. First, more attention is paid to the role of mathematics as a part of STEM (Science, Technology, Engineering and Mathematics). Second, a new term has been introduced in the mathematics curricula: ‘computational thinking’. In the literature on mathematics and technology education, this idea has been around for 16 years (Wing, 2006), but it is now making its appearance in our mathematics curricula. This leads to new chapters in mathematics text books: logic and electronic gates, graph theory, algorithms and programming, and for pupils in some economics classes also linear programming.

I find many of these topics interesting, especially graph theory. But new topics always come at the expense of other topics. What is being covered less? Alas, geometry (e.g. inscribed angles in circles). And the history of mathematics is not even mentioned in the new curricula, nor the link between mathematics and art.

These trends also affect my personal work situation. In my secondary school in Brussels, I teach according to the new curricula. In teacher training, I still teach strong geometry courses to future teachers, but they will have to teach less geometry. My history of mathematics course, in which students developed workshops for colleagues, has regrettably been dropped. The curriculum makers’ motivation is that the new topics are important in this computer age: “Algorithms are becoming important because there are computers everywhere, even in pupils’ school bags.”

This brings me to the question: is it true that algorithms and computational thinking are typically associated with computers? Algorithms *for* computers? Or are algorithms of all times? Algorithms *before* computers? Students and young colleagues mostly see algorithms as something to feed computers with. Let’s look to some examples of algorithms, using material developed by my students.

2 Algorithms for (square) roots

How to calculate a root? If you ask a pupil, the predictable answer is: with a calculator. But there have not always been calculators. Another way is with your naked hands. In Dutch, ‘wortel’ (root) is the same word as ‘carrot’. The calculation of roots is ‘the extraction of carrots’. Two students of mine made a workshop about this. As a joke, participants who calculated the roots correctly were allowed to pull real carrots from a tub of potting soil and take them home.

We will look in detail at two historical algorithms for calculating square roots: the Babylonian and the Chinese. There were also Indian ones.

2.1 Babylonian algorithm

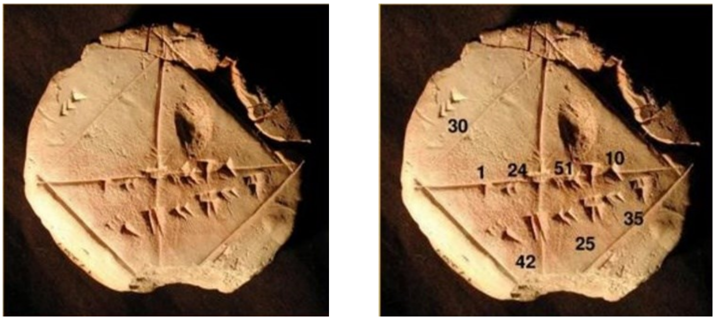


Figure 1. Tablet YBC 7289

Figure 1 shows a tablet from about 4,000 years ago (Yale Babylonian Collection 7289, USA). It testifies that the Babylonians of the time had an algorithm for calculating square roots. According to Fowler and Robson (1998), this was a hand-held tablet on which a student solved school problems. It could be erased and reused. Let us call it the smartphone of the Old Babylonian age.

Babylonians used a numeral system with base sixty, or ‘sexagesimal’ numbers. This is a very convenient numeral system because 60 has a large number of divisors.

On one of the diagonals, we can read the sexagesimal number $1;24;51;10$, which means $1+\frac{24}{60}+\frac{51}{60^2}+\frac{10}{60^3}\approx 1.41421296$, a good approximation for $\sqrt{2}\approx 1.41421356$. Above, we read the number 30 and under the previous number, the sexagesimal number $42;25;35$, which equals $42+\frac{25}{60}+\frac{35}{60^2}\approx 42.42638889$, an approximation for $30\sqrt{2}\approx$

42.42640687. So, the pupil multiplied the side 30 of the square with $\sqrt{2}$ to determine its diagonal.

Another clay tablet (Vorderasiatische Abteilung Tontafeln 6598, Berlin, Germany) presents the following problem: "Find the length of the diagonal of a door 10 units wide and 40 units high." According to the (later) Pythagorean theorem, this amounts to: find $\sqrt{1700}$.

According to Swerts (2012) and others, the method used by the Babylonians for calculating $\sqrt{1700}$ is essentially the same as that of the later Heron of Alexandria (1st century AD). We make a guess, e.g. 40. This guess serves as a first approximation. The second approximation is then the average of 40 and $\frac{1700}{40}$. If we stop here we find 41.25, written decimally, or 41;15 in sexagesimal notation. We can continue, taking as a third approximation the average of 41.25 and $\frac{1700}{41.25}$ and so on.

If we interpret this algorithm geometrically, it becomes much clearer, and I would not be surprised if the Babylonians invented it by thinking geometrically. The problem is: "Given a square with area 1700, find the side." The guess 40 is too small. If we make a rectangle with area 1700 and that side 40, the other side $\frac{1700}{40}$ is too large because the area is still 1700 (Figure 2). The side of the square must be somewhere in between. And the simplest number between two numbers is the average.

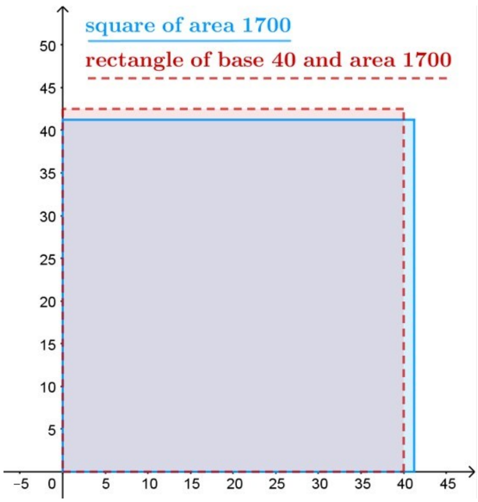


Figure 2. Geometric interpretation of the Babylonian algorithm

Note that this ancient method is equivalent to Newton's method. The approximation of $\sqrt{1+x}$ by replacing the graph of the square root function by its tangent in $\mathrm{x}=1$ gives $\sqrt{1+x} \approx 1+\frac{1}{2}\mathrm{x}$ and this is the average of the first guess 1 and the second guess $1+x$ (Figure 3). Applied to our example of the door, this gives:

$$\sqrt{1700} = \sqrt{1600+100} = 40\sqrt{1+\frac{1}{16}} \approx 40\left(1+\frac{1}{32}\right) = 41.25.$$

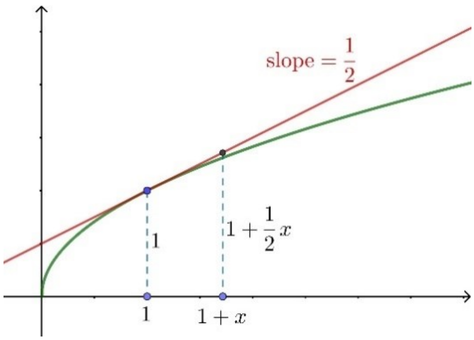


Figure 3. Newton's method

2.2 Calculating the decimal digits of a square root

When I was in primary school in the 1960s and 1970s, we studied an algorithm to calculate the decimal digits of a square root one by one. This algorithm originates from the Ancient Chinese. You can find an example in chapter 4 of the Nine Chapters of Mathematics (Jiuzhang Swanshu, 10th to 2nd century BC). The method is called *Kai Fang*, opening the square (Burgos & Beltrán-Pellicer, 2018). In the 3rd century AD, Liu Hui gives a detailed description of the method. Again, I am convinced that the algorithm is geometrically inspired.

Let's look at an example: $\sqrt{7356\,796}$. This is not the original example of the Ancient Chinese; it is the example of my students Vankriekelsvenne and Vanmarsenille (2003). We invite the reader to follow the steps in Figure 4, starting with a square determined by the first digit and adding *gnomons* in order to find the next digits of the square root. A gnomon is a figure to be added to a square to make a larger square. A gnomon can be transformed into a long rectangle.

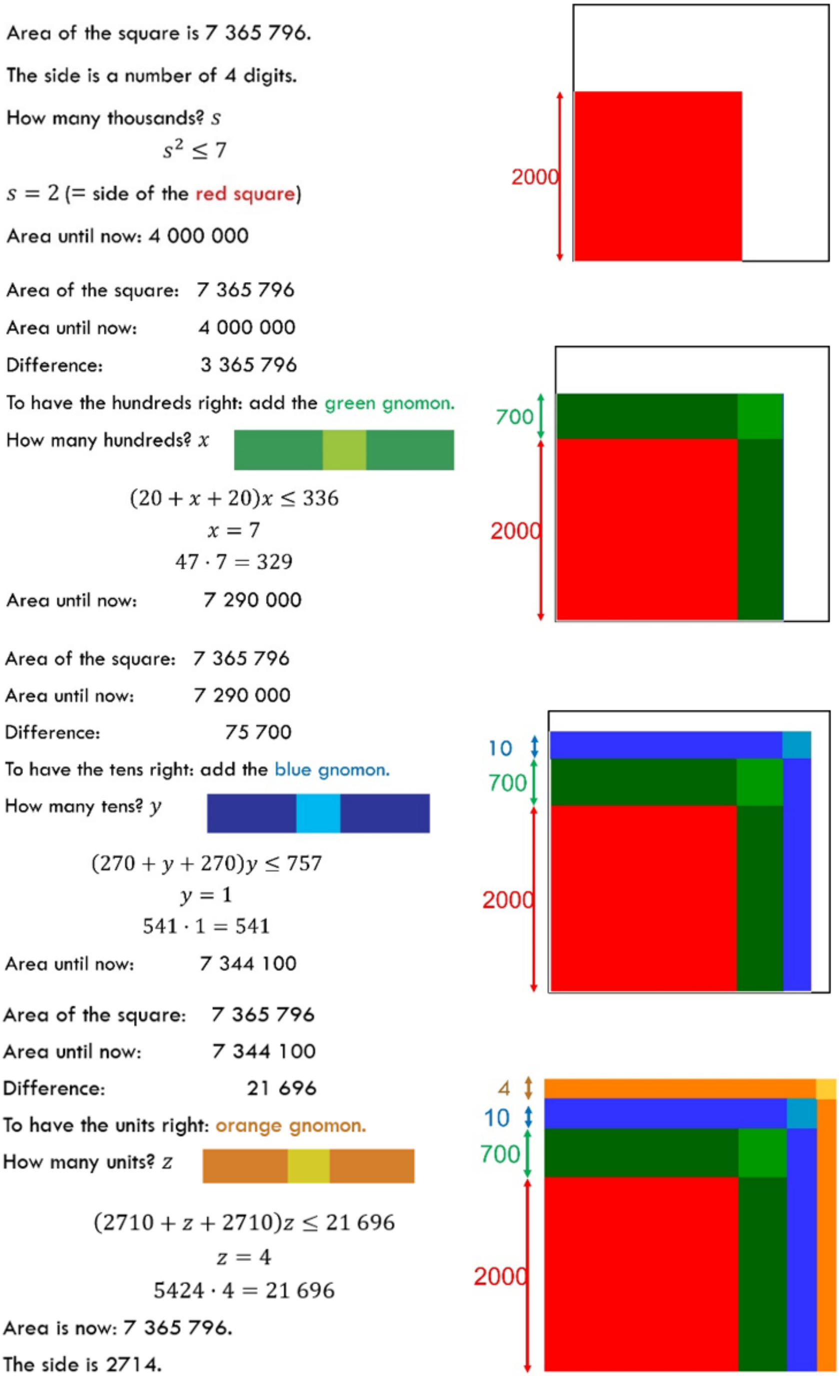


Figure 4. Geometrical search for the digits of the square root

At school, I learned the algorithm without the geometric explanation. Figure 5 shows the calculation for the same example as in Figure 4, with the same steps. Next to it, the figure shows the calculation, with the same algorithm of the first digits of $\sqrt{2}$.

$$
\begin{array}{rr}
 & \sqrt{7\,36\,57\,96} = 2714 \\
2^2 = & \underline{4} \\
 & 3\,36 \\
47 \cdot 7 = & \underline{3\,29} \\
 & 7\,57 \\
541 \cdot 1 = & \underline{541} \\
 & 216\;96 \\
5424 \cdot 4 = & \underline{216\;96} \\
 & 0
\end{array}
\qquad
\begin{array}{rr}
 & \sqrt{2,00\,00\,00} = 1{,}414\ldots \\
1^2 = & \underline{1} \\
 & 1\,00 \\
24 \cdot 4 = & \underline{0\,96} \\
 & 4\,00 \\
281 \cdot 1 = & \underline{2\,81} \\
 & 119\;00 \\
2824 \cdot 4 = & \underline{112\;96} \\
 & 6\,04 \\
 & \cdots
\end{array}
$$

Figure 5. Same algorithm, as we wrote it down at school

There is a similar algorithm to compute the digits of the cubic root of a number, also inspired by geometry, starting with a cube and adding solid gnomons. For more details, see TwoPi (2008).

2.3 Roots of quadratic equations

In the 9th century, the Persian scholar Al Khwarizmi gave a systematic algorithm for the solution of the six types of quadratic equations. In fact, the word ‘algorithm’ is derived from his name. He had to distinguish different types because the numbers had to be positive. A negative term of a modern equation appeared on the other side of the equal sign in Al Khwarizmi’s equation. For example, one of the types was $ax^2 = bx + c$, with a, b and c positive.

The solution recipe was explained geometrically. In figure 6, this is illustrated with the equation $x^2 = 3x + 4$ (Roelens & Van den Broeck, 2015). Al Khwarizmi argues that geometric visualization helps to understand his algebraic algorithms (quoted by Siu, 2002): “*We have now explained these things concisely by geometry in order that what is necessary for an understanding of this branch of study might be made easier. The things which with some difficulty are conceived by the eye of the mind are made clear by geometric figures*.”

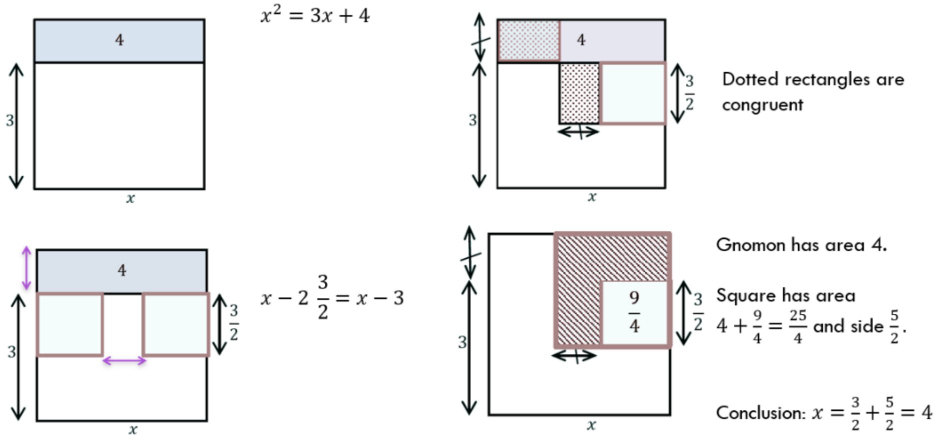


Figure 6. Al Khwarizmi's geometric solution of the equation $x^2 = 3x + 4$

2.4 Conclusion so far

The algorithms to calculate square roots and solutions of quadratic equations are much older than electronic computers. They are largely inspired by geometry and it is through geometry that they can be understood. The practical *use* of the algorithm to calculate the digits of a square root, has been rendered obsolete by the advent of computers and calculators. Still, we think that the geometric discovery, guided reinvention and explanation of the algorithm can be an interesting activity for pupils.

3 Euler graphs

Let's move on to another branch of mathematics. Graph theory has now been added to the mathematics curriculum in Flemish secondary schools because the internet, Facebook, ... are all big graphs, as is the road network in which the GPS (Global Positioning System) has to find the shortest or fastest routes. Graph theory is really something of the computer age...

Again: this theory may not date back to the Babylonians, but it certainly predates computers.

3.1 Euler's theorem

The origins of graphs are usually attributed to Leonhard Euler (18th century), the incredibly versatile Swiss mathematician. Euler wrote so much that, according to the Dutch Wikipedia, it would take an estimated 50 years to transcribe all his works by hand at a rate of eight hours of writing a day.

In Euler's time, there were seven bridges across the river Pregel connecting the various parts of the Prussian city Königsberg (figure 7). After World War II, Königsberg was added to Russia as an exclave between Poland and Lithuania, and has since then been called Kaliningrad.

Euler posed the question whether it was possible to make a walk in Königsberg that passes precisely once over each bridge and returns to the starting point.

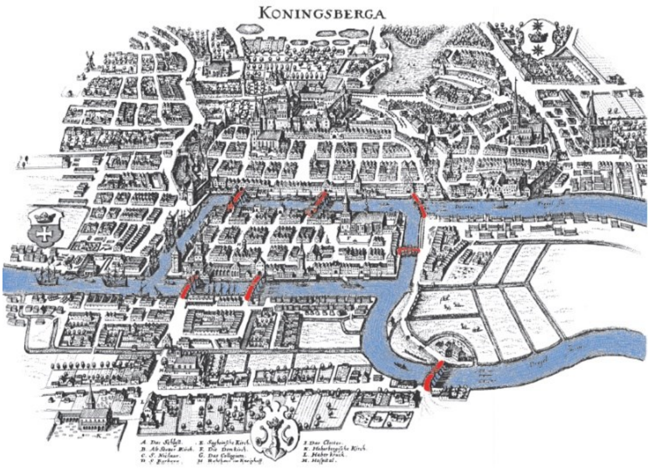


Figure 7. The seven bridges of Königsberg

The solution with graph theory has become a classic. The parts of the city separated by the river are represented by vertices and the bridges by edges connecting these vertices (Figure 8). It is not a single graph but a multigraph, because a pair of vertices can be connected by more than one edge. A walk that goes precisely once over each edge and returns to the starting vertex is called a Eulerian circuit. The degree of a vertex is the number of edges adjacent to that vertex. To have a Eulerian circuit, the degrees of all vertices must be even. In the multigraph represented by

Königsberg, the degrees of vertices a, b, c and d are 5, 3, 3 and 3, respectively. So the requested walk is not possible.

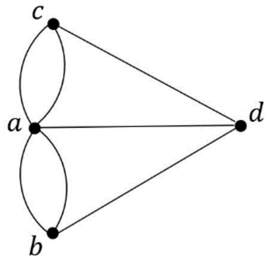
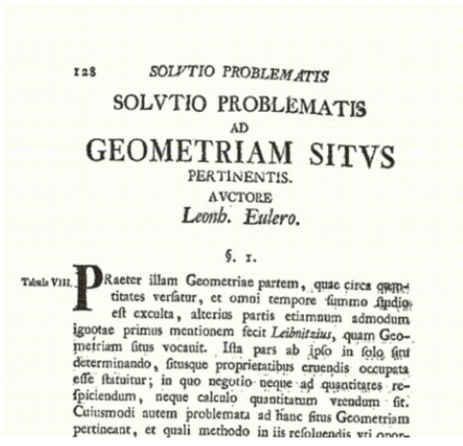


Figure 8. Graph representation of Königsberg

I was convinced that this solution was Euler’s and that Euler effectively used a graph with vertices and edges. When I read Euler’s text (Euler, 1741, Figure 9), I was surprised that Euler did not actually draw a graph, but continued to reason with city regions and bridges. He worked on his simplified drawing of the city, figure 10.



128 SOLVTIO PROBLEMATIS

SOLVTIO PROBLEMATIS
AD
GEOMETRIAM SITVS
PERTINENTIS.
AVCTORE
Leonh. Eulero.

§. 1.

Tabula VIII. PRaeter illam Geometriae partem, quae circa quantitates verſatur, et omni tempore ſummo ſtudio eſt exculta, alterius partis etiamnum admodum ignotae primus mentionem fecit *Leibnitzius*, quam Geometriam ſitus vocauit. Iſta pars ab ipſo in ſolo ſitu determinando, ſitusque proprietatibus eruendis occupata eſſe ſtatuitur; in quo negotio neque ad quantitates reſpiciendum, neque calculo quantitatum vtendum ſit. Cuiusmodi autem problemata ad hanc ſitus Geometriam pertineant, et quali methodo in iis reſoluendis vti opor-

Figure 9. Euler’s text about the bridges of Köningsberg

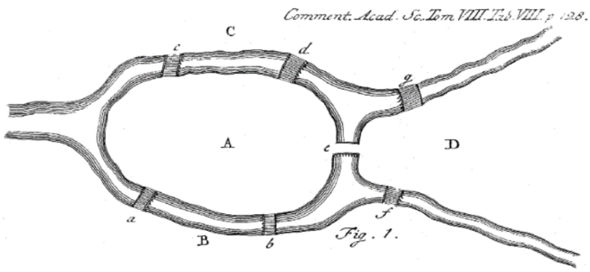


Figure 10. Euler's drawing of the city of Königsberg

"[The problem] could be solved by a complete enumeration of all the walks" he writes, but he prefers a much simpler method. He codes each walk by a sequence of letters, e.g. ABA means start in A, take a bridge to go to B, return to A with another bridge. He proves the impossibility of the requested walk by reasoning on the basis of the number of occurring letters in these codes. For more details, see Barnett (2009) or the original article Euler (1741). Ultimately, he concludes that the number of bridges for each area must be even to make a requested walk possible. He ends by remarking: "The question remains how the walk is to be carried out."

Euler's theorem, in modern terms, is the implication: "*If in a connected multigraph there is a Eulerian circuit, then all vertices have even degree.*" His other theorem is about a Eulerian *trail*, a walk going precisely once over each edge but not returning to the starting vertex: "*If in a connected multigraph there is a Eulerian trail, then exactly two vertices have odd degree.*"

His last comment can be understood as admitting that he does not actually prove the converse theorems. This will be done more than a century later by Hierholzer.

3.2 Hierholzer's algorithm

Carl Hierholzer (1873), in a short article, proves the converse theorems: "*In a connected multigraph: if all vertices have even degree, then there is a Eulerian circuit; if exactly two vertices have odd degree, then there is a Eulerian trail.*"

He repeats the theorems and proofs of Euler, but in a version with vertices and edges, much easier to read than Euler's text. Then he proves his theorems, the converse theorems of Euler's, by describing an algorithm for constructing a Eulerian circuit or trail.

Let's illustrate Hierholzer's algorithm with an example. Take the multigraph with vertices $A, B \dots J$ of Figure 11. The degrees of the vertices are even, so we should be able to make a Eulerian circuit. Start somewhere and walk on the edges, without repeating the same edge, until you get stuck. For example, start in A and make the walk $ABCDAIJJA$. Colour the edges you walk on in a first colour, e.g. red. Because all degrees are even, you cannot get stuck elsewhere than in the starting vertex A. The walk had to be a sub*circuit*. If all edges were coloured, you would have finished a Eulerian circuit, but in the example this is not the case. Now, start at a vertex where some but not all edges are coloured, e.g. B, and make a new subcircuit on non-coloured edges until you get stuck again, for example $BDEGIDB$ (green). Repeat this procedure, e.g. make subcircuit $EFGHIE$ (blue). Now in the example all edges are coloured. To make a Eulerian circuit from our subcircuits, you replace B in the first subcircuit by the second subcircuit: $A\ BDEGIDB\ CDAIJJA$ and then you replace E in the second subcircuit by the third subcircuit: $A\ BD\ EFGHIE\ GIDB\ CDAIJJA$. This is a Eulerian circuit in the given multigraph.

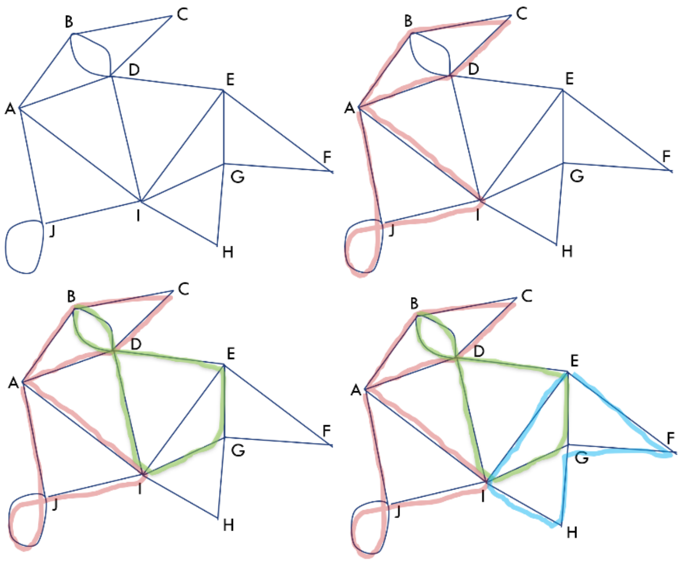


Figure 11. Example of Hierholzer's algorithm

4 Algorithms before computers

4.1 No mathematics without algorithms

Long before there were electronic computers, algorithms formed an important aspect of mathematics.

In section 2, we discussed geometrically underpinned Babylonian and Old Chinese algorithms to calculate square roots and a still-used medieval Persian algorithm to solve a second degree equation.

There are many other examples. Euclid (300 BC) devised an algorithm for the greatest common divisor of two natural numbers. Any Greek construction with ruler and compasses can be viewed as an algorithm in which, starting from a finite number of given points, circles and lines are added and intersected step by step to obtain the desired result. The Chinese mathematician Liu Hui (3rd century) systematically solved systems of first-degree equations using what later came to be called Gauss's pivot method (19th century).

Many proofs also contain an algorithm. In section 3 we explained Hierholzer's proof of the existence of a Eulerian circuit in a multigraph in which the degrees of all vertices are even. Hierholzer proves this by giving an algorithm to construct such a Eulerian circuit. When Euclid proves that there are infinitely many prime numbers, he does so by describing an algorithm to produce, for any list of prime numbers, an additional prime number missing from the list.

Man-Keung Siu (Siu, 2002) discusses the distinction between 'algorithmic' and 'dialectic' mathematics. Algorithmic mathematics is about finding solutions, using fixed methods and algorithms. Dialectic mathematics is about explaining and proving. When I read Siu's article, I was thinking that the recent trend in our curricula is towards more algorithmic mathematics and less dialectical mathematics. On the other hand, the examples I elaborated or cited above show that dialectic and algorithmic mathematics are inseparable and intertwined. This is also Man-Keung Siu's conclusion: they are two aspects of a same reality, like Yin and Yang in the Chinese tradition: "*In the teaching of mathematics we should not just emphasize one at the expense of the other. When we learn something new we need first to get acquainted with the new thing and to acquire sufficient feeling for it. A procedural approach helps us to prepare more solid ground to build up subsequent conceptual understanding. In turn, when we understand the concept better we will be able to handle the algorithm with more facility*." (Siu, 2002)

4.2 Inventing and explaining algorithms geometrically

So, no mathematics without algorithms. But the essence of mathematics is not the execution of algorithms. In some first-grade mathematics textbooks, many solution procedures are laid down in 'step-by-step plans' that students have to perform literally. In part, of course, it is necessary for pupils to automate certain calculations or solution methods so that they do not have to think about them from scratch every time they need them. But let's not overdo it: for executing algorithms, there are now computers, and pupils are no computers. Running an algorithm is less interesting than searching for solutions and explanations, coming up with algorithms...

Let us not turn mathematics lessons into merely 'applying algorithms'. Let's teach about algorithms, their history, their geometric inspiration. Let's give pupils the opportunity and the time to invent algorithms.

4.3 More or less algorithms in our computer era?

Obviously, computers have made algorithms *more relevant*. Algorithms and computer programmes play a big role in the background of the internet, social media, search engines, navigation systems... They determine our lives and sometimes threaten our privacy.

On the other hand, computers have made algorithms *less relevant*. In my own job as a mathematics teacher, I had to program much more often in the previous century than in this one. This is because many things are now pre-programmed, in GeoGebra, in graphing calculators, in all kinds of apps. Having a graph drawn by a computer, investigating the effect of parameters: I remember that this required programming. In the last 25 years, I hardly ever had to program. And now it will be necessary again, because the curriculum makers consider it part of the computer age.

I want to end this article with some quotes from Donald Knuth. He is 84 now. He is considered the father of programming and the inventor of TEX. He also wrote about the history of Babylonian algorithms (Knuth, 1972).

"*Programming is the art of telling another human being what one wants the computer to do.*"

"*An algorithm must be seen to be believed*."

"*Programs are meant to be read by humans and only incidentally for computers to execute.*" (Knuth, 1969)

REFERENCES

Barnett, J.H. (2009). Early Writings on Graph Theory: Euler Circuits and The Königsberg Bridge Problem. *Resources for Teaching Discrete Mathematics. MAA Notes #74* (pp. 197-208). The Mathematics Association of America. https://www.maa.org/sites/default/files/pdf/pubs/books/members/NTE74.pdf

Burgos, M., Beltrán-Pellicer, P. (2018). On Squares, Rectangles and Square Roots. *Convergence 15.*

Chabert, J.-L, Barbin, E., Guillemot, M. (1995). *Histoire d'algorithmes: du caillou à la puce.* Paris: Belin.

Euler, L. (1741). Solutio problematis ad geometriam situs pertinentis. *Euler Archive - All Works. 53* https://scholarlycommons.pacific.edu/euler-works/53 (Latin text). https://www.cantab.net/users/michael.behrend/repubs/maze_maths/pages/euler_en.html (English translation).

Fowler, D., Robson, E. (1998). Square Root Approximations in Old Babylonian Mathematics: YBC 7289 in Context. *Historia Mathematica 25*.

Goldberg, D. (2000). Ode to the Square Root: A Historical Journey. *Humanistic Mathematics Network Journal 23*.

Hierholzer, C. Wiener, C. (1871). Ueber die Möglichkeit, einen Linienzug ohne Wiederholung und ohne Unterbrechung zu umfahren. *Mathematische Annalen VI*, 30–32. https://eudml.org/doc/156599

Kindt, M. (2007). *√ met een staart*. Non-published hand-out Nationale WiskundeDagen.

Kindt, M. (2022). Algoritmen met een staart. *Euclides* 98(2) *and 98*(3).

Knuth, D.E. (1968). *The Art of Computer Programming* (3rd ed.). Addison Wesley.

Knuth, D.E. (1972). Ancient Babylonian Algorithms. *Communications of the ACM* 15(7), pp. 671-677.

'Leonhard Euler.' (2022, December 22) In *Wikipedia*. https://nl.wikipedia.org/wiki/Leonhard_Euler

Moore, C., Mertens, S. (2012). *The Nature of Computation.* Oxford University Press.

Moyon, M. (2007). La tradition algébrique arabe d'al Kwarizmi au Moyen Age latin et la place de la géométrie. Barbin, E., Bénard, D. (Ed.) *Histoire et enseignement des mathématiques: rigueurs, erreurs, raisonnements* (pp. 289-318).Clermont-Ferrand: Institut National de Recherche Pédagogique.

Perucca, A. (2020). De zeven bruggen van Koningsbergen. *Uitwiskeling 36*(1).

Roelens, M., Van den Broeck, L. (2015). Veeltermvergelijkingen vroeger en nu. *Uitwiskeling 31*(4).

Roelens, M., Vanlommel, E. (2020). Redeneren en puzzelen met grafen. *Uitwiskeling 36*(2).

Siu, M-K (2002). *'Algorithmic mathematics' and 'dialectic mathematics': the 'yin' and the 'yang' in Mathematics Education.* Text of plenary lecture at the 2nd International Conference on the Teaching of Undergraduate Mathematics, Crete, July 2002. https://hkumath.hku.hk/~mks/AlgorithmicDialectic.pdf

Swetz, F. (2012). *Mathematical Expeditions. Exploring Word Problems across the Ages.* Baltimore: The John Hopkins University Press.

Vankriekelsvenne, K., Vanmarsenille, J. (2003). *Handmatig worteltrekken*. Diepenbeek: non-published workshop material.

Tycho's Wiskunde: [Handmatig worteltrekken] Deel 2 - Decimaal voor decimaal. https://www.youtube.com/watch?v=JjUJua1Gptg

Tycho's Wiskunde: [Handmatig worteltrekken] Deel 6 – Derdemachtswortels. https://www.youtube.com/watch?v=gpma8dhD69Q

TwoPi (2008). *Root extraction, part II: cube roots*. https:/threesixty360.wordpress.com/2008/02/11/root-extraction-part-ii-cube-roots/

Wing, J.M. (2006). Computational Thinking. *Communication of the ACM*.