

Computational complexity theory

Million Dollar Algorithm?



MAREK KUBALE

Faculty of Electronics, Telecommunications,
and Informatics
Gdańsk University of Technology
Member of the PAS Committee on Computer Science,
kubale@eti.pg.gda.pl
Prof. Marek Kubale is a professor at the Department of
Algorithms and Systems at Gdańsk University of Technology

Computers have already become nearly ubiquitous in our daily lives. Yet it takes not just hardware, but also software to make a computer. Some discoveries in the programming field have had a direct impact on our standard of living

The basic concept in computer programming is the notion of algorithm. The word is derived from the nickname of Muhammad ibn Musa, a medieval Persian mathematician, who was called al-Khwarizmi in Arabic after his family home, and that in turn became *Algorismus* in Latin. For centuries there was no formal definition, but descriptions of procedures to solve various problems (even including dietary ones) had been written down since antiquity. These days such descriptions are called algorithms. Today we would define an algorithm as an unambiguous way of processing certain input data into certain output data within a finite amount of time.

2300 years of algorithm studies

The first nontrivial algorithm is considered to be Euclid's method for computing the greatest common divisor of two natural numbers. Although it was known previously, it was first written down by Euclid in his *Elements* around 300 BC. The second algorithm worth mentioning was a method used around 100 AD by Chinese generals to check the number of soldiers. Now known as the Chinese Remainder Theorem, this algorithm has many more applications today.

In the late 19th and early 20th centuries, mathematicians grew interested in general questions: what can be calculated, what functions are computable, for what problems algorithms exist, and more broadly – whether all mathematical theorems can be proven either true or false. In 1900, the great German mathematician David Hilbert formulated 23 great challenges for mathematicians, with number ten being the question of whether there exists an algorithm to find an integer solution for a given polynomial equation with integer coefficients. It was only 70 years later that the Russian mathematician Yuri Matiyasevich proved that the answer to this question is no. Hilbert's tenth problem touched off enormous interest in computability – a field that seeks answers to the question of whether a

Computers are becoming nearly ubiquitous in our daily lives





Marek Kuballe

A motif inspired by Regular Space Division III by M. C. Escher – an artist's way of solving the tiling problem

given problem has a solution in the form of an algorithm or not.

In 1936 the British mathematician Alan Turing published a paper "On Computable Numbers, with an Application to the *Entscheidungsproblem*" (using a German term meaning "solution problem"), in which he answered the most important question mathematicians and philosophers of mathematics had been asking themselves in the early 20th century: Can a machine be constructed to solve mathematical theorems automatically, without human involvement? He showed that such a machine could never be built, and in his proof he used the concept of an abstract computing device now known as a Turing Machine. Today computer science students study Turing Machines as a theoretical model of the computer, which was actually invented a few years later.

The first digital electronic machines (computers) were constructed during WWII. They were of sizeable dimensions and their computational power was billions of times smaller than that of today's PCs, but they imparted new momentum to the study of algorithms. All domains of science have benefitted from this new invention, especially combinatorics and graph theory.

The year 1971 marked a breakthrough in the study of algorithms, when the American computer scientist Steven Cook showed

that the problem of the satisfiability of logical formulas is computationally difficult to solve. By so doing he laid the foundations of a very important class of problems, called NP-hard. Aside from several thousand combinatorial problems, this class also includes the "travelling salesman problem" and issues in code-breaking.

Seven millennium problems

In the year 2000, scientists from Clay Mathematics Institute in Massachusetts, taking their cue from Hilbert's idea 100 years before, formulated 7 open problems that had so far resisted solution. A prize of \$1 million was set for the solution of each of these new "millennium problems." The list is topped by the algorithmic question "P=NP?" To simplify things greatly, the issue here is that we do not know whether certain difficult computational problems, such as the travelling salesman problem, cannot be resolved because no algorithmic solution exists (then $P \neq NP$), or whether their solution is in fact possible but mankind has so far lacked the ingenuity to come up with such an efficient algorithm (then $P = NP$). Today only 6 of the 7 millennium prize problems remain open, since one of them (the Poincaré Conjecture) was solved in 2003. The \$1 million prize was not paid out, however, since the author of the solu-

Computational complexity theory

tion, the Russian mathematician Grigoriy Perelman, refuses not only to accept any prizes, but also to have any contact with the media whatsoever.

Types of problems

Saying that a problem is solvable using an algorithm means that a computer program can be written to yield a correct answer for any appropriate set of input data, in finite time, assuming access to unlimited memory. The requirement of time-effectiveness is very important, and on its basis problems can be classified into five groups.

The first consists of non-algorithmic problems, which cannot be resolved using computer programs. One example is the tiling problem, which means deciding whether identical copies of a given polygon can be used to fully cover an infinite plane. The existence of non-deterministic problems is proof that the human mind can do something more than computers can – it can work non-deterministically. This means it is not possible to create an artificial intelligence equaling the intelligence that humans are endowed with. The second group

consists of possibly non-algorithmic problems, for which no finite-time algorithm has yet been devised, but neither is there any proof that no such algorithm exists.

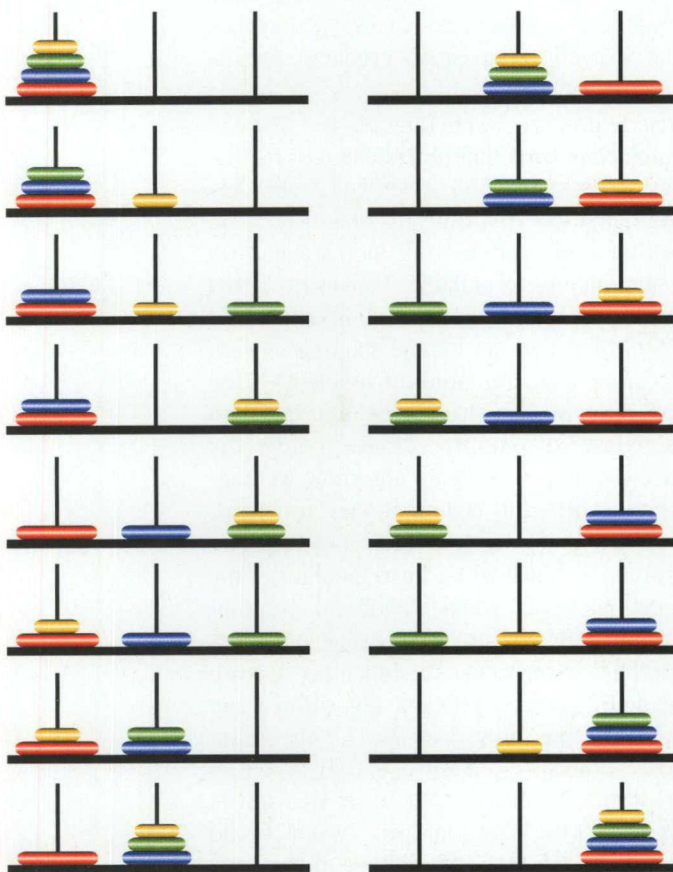
The third category consists of exponential problems, for which the time required to find a solution rises exponentially with the size of the task. One example is the famous Towers of Hanoi puzzle, familiar to preschoolers everywhere as a block-stacking toy. The problem is to move all the disks from the first rod to the third, moving one at a time, while never allowing a larger disk to rest upon a smaller one. Solving this puzzle takes 2^{n-1} steps. A related legend states that Buddhist monks at a certain monastery in Hanoi have been moving 64 golden disks at a rate of one disk per second, and the end of the world will come when they move the last disk into place. So how much time is left until the end of the world?

The fourth class consists of possibly exponential problems, for which no polynomial-time algorithm has yet been discovered, but for which it has also never been proven that no such algorithm exists. One example is factorization, the problem of decomposing a given number into its prime factors. Lastly, the fifth category consists of polynomial-time problems, for which there exist algorithms that take polynomial runtime – such as various types of sorting, which is a very important algorithmic problem. Someone once remarked that half of the world's computers at any given moment are busy performing sorting operations.

Optimal planning

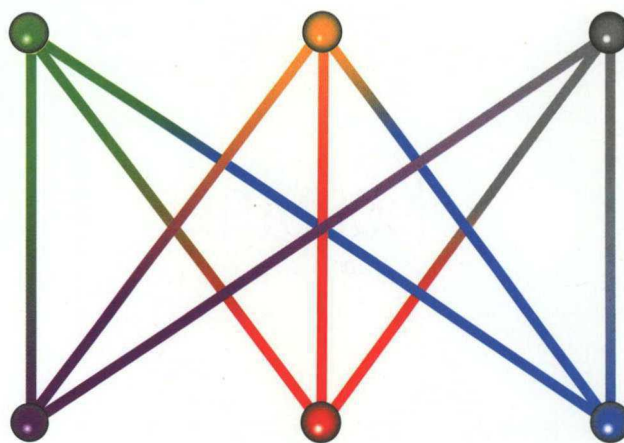
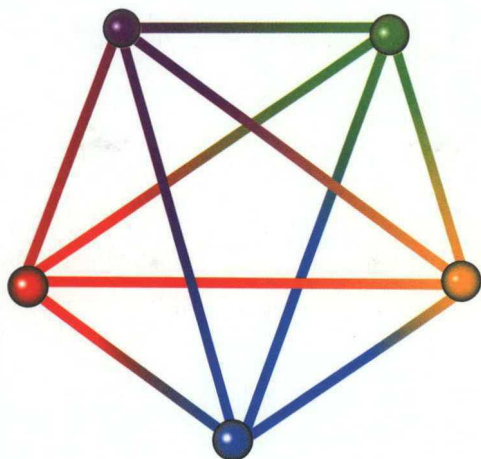
The history of algorithmics has been full of spectacular algorithm ideas. One rich field is linear programming, which is important in view of its numerous practical applications in production planning, resource allocation, task scheduling, and transportation problems. In the WWII years, the problem of planning an optimal diet arose. Food products contain nutrients and vitamins in various proportions. If we know the magnitude of the supplies of each product and their prices, we can define a way to satisfy a set of nutritional needs at minimal cost. In 1947 the American George Dantzig developed the "simplex method" for resolving this problem. Its name is derived from the simplex, a convex shape that is a multidimensional generalization of the triangle. Theoretically Dantzig's method works in exponential time,

The towers of Hanoi – a preschooler toy or a serious algorithmic problem? The task is to move all the disks from the first rod to the third, one at a time, while never resting a larger disk on a smaller one



Paweł Adamów based on <http://mathworld.wolfram.com/>

Paweł Adamów based on Marek Kubale



but in practice it is very efficient. Dantzig gave an example of assigning 70 workers to 70 workstations, a problem with $70!$ (70 factorial) permissible solutions. That is an enormous number, but his algorithm provides an optimal solution almost immediately.

In 1979, Leonid Khachiyan, a mathematician of Armenian descent, published the so-called "ellipsoidal method" for linear programming, which was the first provably polynomial algorithm. Khachiyan's result is mainly of theoretical importance, since it is estimated that its advantage over the simplex method becomes manifest only with upwards of 1000 restrictions and $n=5000$ variables, but it still marked a breakthrough in linear programming. Finally, in 1984, the Indian mathematician Narendra Karmarkar developed the "interior point method." A computer running this algorithm has to perform around $l \cdot n^{3.5}$ floating-point operations, where l is the number of bits required to record data values. This is currently the fastest-known asymptotic algorithm for linear programming.

Banks and graphs

Another problem is testing primality (whether a given number is prime), which is important in cryptography for banking and military applications, telecommunications, etc. In 2002, this problem that had been intriguing humanity since antiquity was solved when the three Indian mathematicians Manindra Agrawal, Neeraj Kayal, and Nitin Saxena published a new method to test a given natural number's primality. Currently, after certain improvements, the "AKS algorithm" named after them can be performed in n^6 time, where n is the number of digits. Despite this success, however, deterministic tests of primality are still significantly slower than probabilistic ones.

Another important issue is known as the graph embedding problem, or finding a way to draw a graph - understood here as a structure

of vertexes (points) connected by edges (lines) - on a flat plane (such as a computer monitor) so that none of its edges cross. Early 20th-century mathematicians were interested in finding out what the necessary and sufficient conditions are for a graph to be planar. This problem was solved in 1930 by the Polish mathematician Kazimierz Kuratowski, who demonstrated that a graph is planar if and only if it does not contain any subgraph homeomorphic to the smallest nonplanar graphs K_5 or $K_{3,3}$.

Of course, working nearly 100 years ago, Kuratowski was not studying algorithms. However, his proof can be used to derive an algorithm of n^6 complexity for testing the planarity of a graph. Nowadays, as a result of work by John Hopcroft and Robert Tarjan in 1974, we can program computers to plot planar graphs in a way that draws out all the symmetries they contain, in time strictly proportional to their size - meaning linear time (which in practice means nearly immediately).

So does $P=NP$?

The most important problem in contemporary theoretical computer science remains the millennium question: $P=NP$? One hundred top theoretical computer scientists were asked in 2002 in which year and with what result they expected the problem to be resolved. Responses were received from 79 professors. Of them, 61 respondents expect that $P \neq NP$, 9 suppose the opposite, and 9 provided a different answer. According to the majority view, we will find out the answer around 2040. ■

Further reading:

- Kubale, M. (2009). *Łagodne wprowadzenie do analizy algorytmów* [An Easy Introduction to Algorithm Analysis]. Gdańsk: Gdańsk University of Technology Publishers.
- Kubale, M. et al. (2004). *Graph Colorings, Contemporary Mathematics* 352. Providence, Rhodes Island: American Mathematical Society.

K_5 and $K_{3,3}$ - the smallest nonplanar graphs. Kazimierz Kuratowski (also jocularly known as K_5 Kazimierz $K_{3,3}$ Kuratowski) showed that every nonplanar graph contains a subgraph homeomorphic to one of them