XI Bienal de Matemática 2024

O Problema do Milênio sobre Intratabilidade Computacional

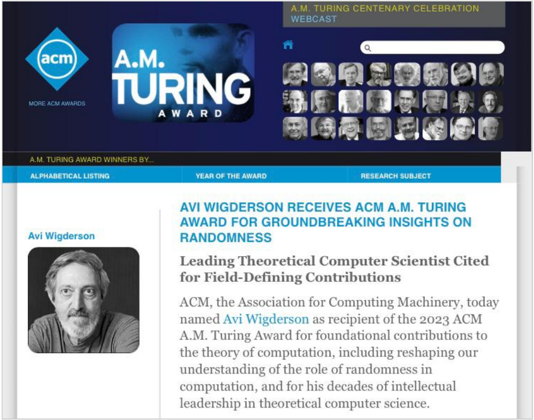
Celina Miraglia Herrera de Figueiredo

COPPE UFRJ

Mathematician wins Turing award for harnessing randomness

Wigderson started exploring the relationship between randomness and computers in the 1980s, before the internet existed, attracted to ideas he worked on by intellectual curiosity, rather than how they might be used

One of the unexpected ways in which his ideas are now widely used was on zero-knowledge proofs, which detail ways of verifying information without revealing the information itself



read Quanta Magazine
watch Zero Knowledge Proof

Abel prize celebrates union of Mathematics and Computer Science

Two pioneers of the theory of computation have won one of the most prestigious honours in mathematics

Since the advent of computers in the twentieth century, the emphasis in research has changed from ‘can an algorithm solve this problem?’ to ‘can an algorithm, at least in principle, solve this problem on an actual computer and in a reasonable time?’

read Abel interview 2021

Today is more difficult to distinguish pure and applied mathematics

Mathematics → Computing
László Lovász (1948, Budapest) grew up a talented child competing at solving hard problems Early inspiration from Paul Erdos, prolific mathematician of the modern era, who focused on the mathematics of discrete objects Interested in basic research as well as in its applications, worked as a full-time researcher at Microsoft for seven years in between academic positions

Computing → Mathematics
Avi Wigderson (1956, Haifa) studied in Israel and the United States and held various academic positions before moving to the IAS in 1999, where he is ever since. Contributed to practically all areas of computer science, in which he attacked any problem with whatever mathematical tools he could find, even from distant fields of study

Abel prize – The Nobel for Mathematics

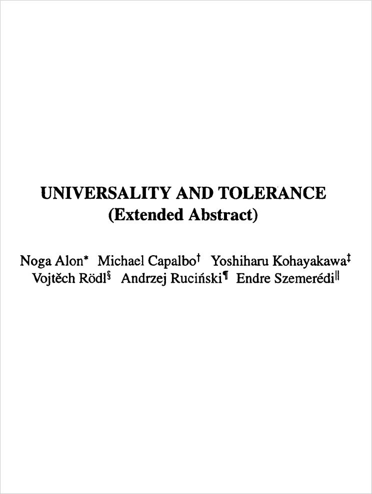
Laureates since 2003 in DM and TCS

2012 Endre Szemerédi – fundamental contributions to discrete math and theoretical computer science

2021 László Lovász and Avi Wigderson – foundational contributions to theoretical computer science and discrete math, and their role in shaping them into central fields of modern mathematics

John Nash awarded Nobel (1994, Game Theory) + Abel (2015, Partial Differential Equations)

The Fields Medal is awarded since 1936 up to four mathematicians under 40 years at the International Mathematical Union Congress, every four years



UNIVERSALITY AND TOLERANCE
(Extended Abstract)

Noga Alon[*] Michael Capalbo[†] Yoshiharu Kohayakawa[‡]
Vojtěch Rödl[§] Andrzej Ruciński[¶] Endre Szemerédi[‖]

Foundations of Computer Science, 2000

Turing award – The Nobel for Computer Science

Laureates since 1966 in theoretical computer science

1974 Donald Knuth – contributions to the analysis of algorithms

1982 Stephen Cook – understanding the complexity of computation

1985 Richard M. Karp – contributions to the theory of algorithms, polynomial-time computability and NP-completeness

1986 Robert Tarjan – design and analysis of algorithms and data structures

INFORMATION PROCESSING LETTERS 2 (1974) 153–157. NORTH-HOLLNAD PUBLISHING COMPANY

A STRUCTURED PROGRAM TO GENERATE ALL TOPOLOGICAL SORTING ARRANGEMENTS

Donald E. KNUTH*
Computer Science Dept., Stanford University, Stanford, Calif., 94305, USA
and
Jayme L. SZWARCFITER **
Universidade Federal do Rio de Janeiro, Argentina

Received 26 October 1973
Revised version received 5 February 1974

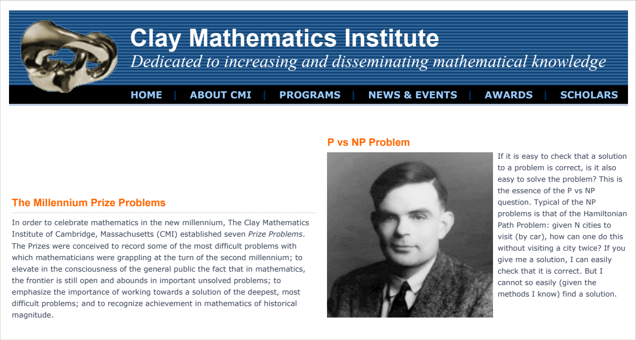
data structures programming languages combinatorial problems

The Millennium Prize Problems

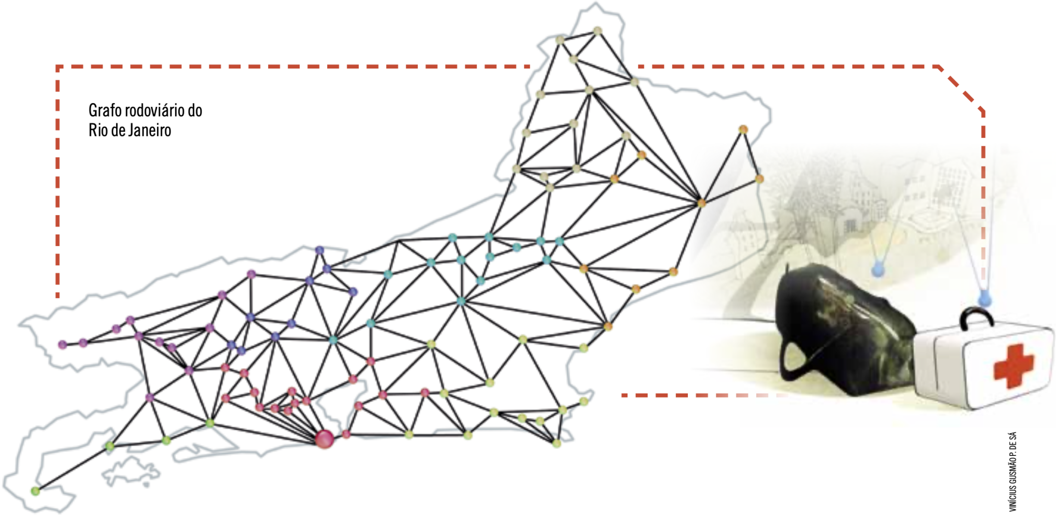
David Hilbert:
23 problems
Paris in 1900

Clay Mathematics Institute:
7 prize problems
Paris in 2000

P versus NP problem has no associated mathematician



watch Vijaya Ramachandran



Grafo rodoviário do Rio de Janeiro

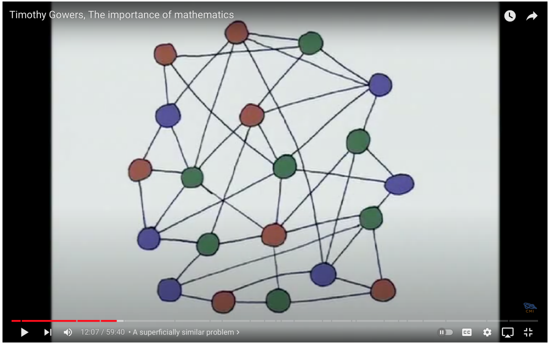
VINÍCIUS GUSMÃO P. DE SÁ

P versus NP – a gift to Mathematics from Computer Science

The question is whether or not, for all problems for which an algorithm can **verify** a given solution quickly (in polynomial time), an algorithm can also **find** that solution quickly

Avi Wigderson expects P not equal NP

Donald Knuth expects P equal NP



Timothy Gowers, The Importance of Mathematics, 2000

watch Donald Knuth: P=NP

Hilbert, Gödel, Turing, von Neumann, Wigderson

Hilbert's two-part dream:
Everything that is true in Mathematics is provable.
Everything that is provable can be automatically computed.

1931 Gödel proved that no matter how hard you try, your set of axioms will always be incomplete, they will not be sufficient to prove all true facts

1936 Turing introduced his Turing machine and proved the unsolvability of the halting problem

1940s–50s Turing and von Neumann played a major role in early development of computers

Kurt Gödel's Letter to John von Neumann - 1956

Princeton, 20 March 1956

Dear Mr. von Neumann:

With the greatest sorrow I have learned of your illness. The news came to me as quite unexpected. Morgenstern already last summer told me of a bout of weakness you once had, but at that time he thought that this was not of any greater significance. As I hear, in the last months you have undergone a radical treatment and I am happy that this treatment was successful as desired, and that you are now doing better. I hope and wish for you that your condition will soon improve even more and that the newest medical discoveries, if possible, will lead to a complete recovery.

Since you now, as I hear, are feeling stronger, I would like to allow myself to write you about a mathematical problem, of which your opinion would very much interest me: One can obviously easily construct a Turing machine, which for every formula F in first order predicate logic and every natural number n, allows one to decide if there is a proof of F of length n (length = number of symbols). Let $\Psi(F, n)$ be the number of steps the machine requires for this and let $\varphi(n) = \max_F \Psi(F, n)$. The question is how fast $\varphi(n)$ grows for an optimal machine. One can show that $\varphi(n) \geq k \cdot n$. If there really were a machine with $\varphi(n) \sim k \cdot n$ (or even $\sim k \cdot n^2$), this would have consequences of the greatest importance. Namely, it would obviously mean that in spite of the undecidability of the Entscheidungsproblem, the mental work of a mathematician concerning Yes-or-No questions could be completely replaced by a machine. After all, one would simply have to choose the natural number n so large that when the machine does not deliver a result, it makes no sense to think more about the problem. Now it seems to me, however, to be completely within the realm of possibility that $\varphi(n)$ grows that slowly. Since it seems that $\varphi(n) \geq k \cdot n$ is the only estimation which one can obtain by a generalization of the proof of the undecidability of the Entscheidungsproblem and after all $\varphi(n) \sim k \cdot n$ (or $\sim k \cdot n^2$) only means that the number of steps as opposed to trial and error can be reduced from N to $\log N$ (or $(\log N)^2$). However, such strong reductions appear in other finite problems, for example in the computation of the quadratic residue symbol using repeated application of the law of reciprocity. It would be interesting to know, for instance, the situation concerning the determination of primality of a number and how strongly in general the number of steps in finite combinatorial problems can be reduced with respect to simple exhaustive search.

I do not know if you have heard that "Post's problem", whether there are degrees of unsolvability among problems of the form $(\exists y)\varphi(y, x)$, where φ is recursive, has been solved in the positive sense by a very young man by the name of Richard Friedberg. The solution is very elegant. Unfortunately, Friedberg does not intend to study mathematics, but rather medicine (apparently under the influence of his father). By the way, what do you think of the attempts to build the foundations of analysis on ramified type theory, which have recently gained momentum? You are probably aware that Paul Lorenzen has pushed ahead with this approach to the theory of Lebesgue measure. However, I believe that in important parts of analysis non-eliminable impredicative proof methods do appear.

I would be very happy to hear something from you personally. Please let me know if there is something that I can do for you. With my best greetings and wishes, as well to your wife,

Sincerely yours,

Kurt Gödel

P.S. I heartily congratulate you on the award that the American government has given to you.

Cook's SAT followed by Karp's 21 problems

1971 Stephen Cook –
SAT NP-complete and
polynomial-time reduction

1972 Richard Karp – Reducibility
among combinatorial problems

Equivalent classic unsolved problems

Either each has polynomial algorithm
or none does

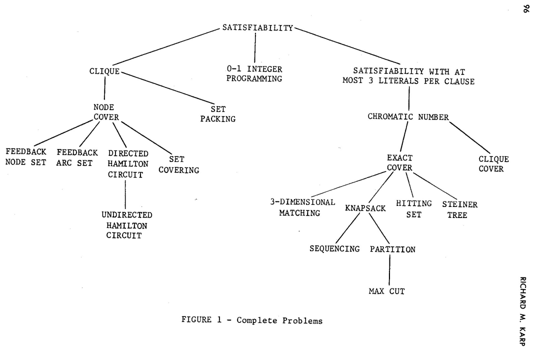


FIGURE 1 - Complete Problems

Knuth's terminology

Problem at least as difficult to solve in polynomial time as those of Cook–Karp class NP

Knuth wrote to 30 people:
Herculean, Formidable or Arduous?

The winning write-in vote is NP-hard, put forward by several people at Bell Labs

SIGACT News 14 January 1974

before looking at the ballots.] It's preposterous to do such a thing in a democracy, but I did it. The resulting weighted average scores were

Herculean	.369
formidable	.373
arduous	.353

In other words, very low. [I'll bet that the term 'polynomial complete' would have fared even worse in the early days; but I'm just trying to heal my wounded feelings when I say this.]

Fortunately, there was a ray of hope remaining, namely the space for write-in votes. I received very many ingenious suggestions; indeed, the write-ins proved conclusively that creative research workers are as full of ideas for new terminology as they are empty of enthusiasm for adopting it.

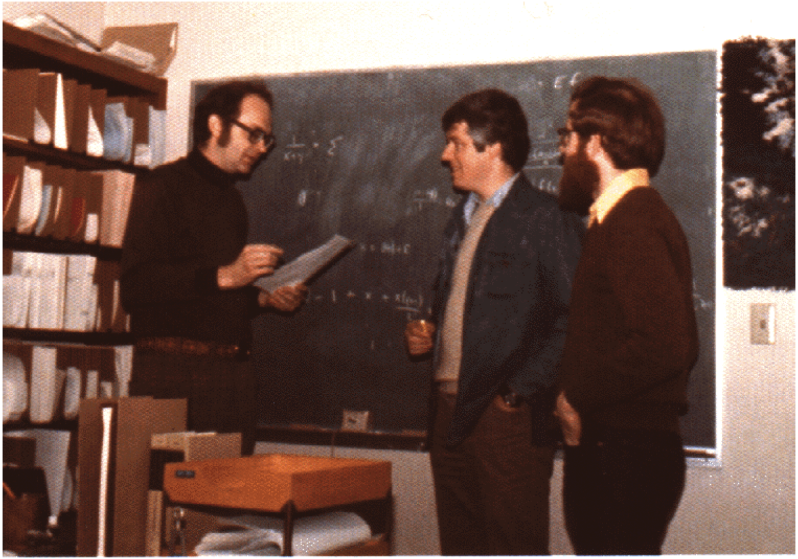
The write-in votes were so interesting, I'd like to discuss them here at some length. First, there were several other English words suggested:

impractical	intractable
bad	costly
heavy	obdurate
tricky	obstinate
intricate	exorbitant
prodigious	interminable
difficult	

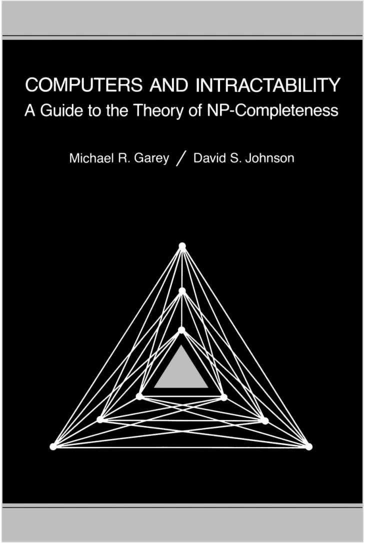
Also, Ken Steiglitz suggested "hard-boiled", in honor of Cook who originated this subject. Al Meyer tried "hard-ass" (hard as satisfiability). [You can see what I mean about creative researchers.]

A terminology proposal, D.E. Knuth, SIGACT News, 1974

Knuth – Garey – Johnson



The Guide is 40 years old



"Despite that 23 years have passed since its publication, I consider Garey and Johnson the single most important book on my office bookshelf. Every computer scientist should have this book on their shelves as well. NP-completeness is the single most important concept to come out of theoretical computer science and no book covers it as well as Garey and Johnson."

Lance Fortnow, "Great Books: Computers and Intractability: A Guide to the Theory of NP-Completeness"

DOI:10.1145/3460351

Advances in algorithms, machine learning, and hardware can help tackle many NP-hard problems once thought impossible.

BY LANCE FORTNOW

COMMUNICATIONS OF THE ACM | JANUARY 2022

Fifty Years of P vs. NP and the Possibility of the Impossible

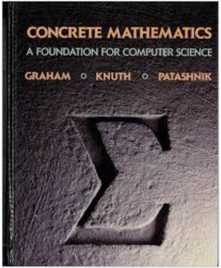
Discrete Mathematics

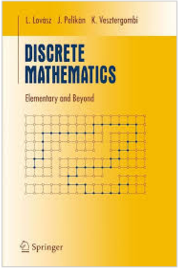
Combinatorics is a branch of mathematics, plays crucial role in computer science, since digital computers manipulate discrete, finite objects

Combinatorial methods give analytical tools for computer algorithms worst-case and expected performance

Concrete Mathematics =
CONtinuous and disCRETE mathematics

a complement to abstract mathematics





Theoretical Computer Science

Studies the power and limitations of computing

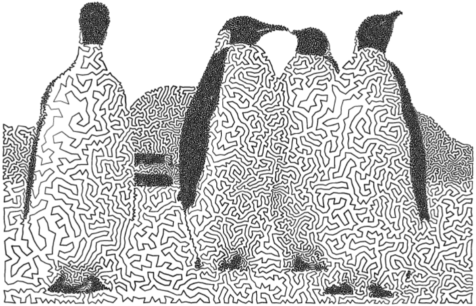
Has two complementary sub-disciplines:

Algorithm Design develops efficient methods for computational problems

Computational Complexity shows limitations on efficiency of algorithms

Discrete mathematics and TCS are allied fields: graphs, strings, permutations are central to TCS

Computing technology is made possible by algorithms, understanding the principles of powerful and efficient algorithms deepens our understanding of computer science, and also of the laws of nature



TSP Art by Craig Kaplan

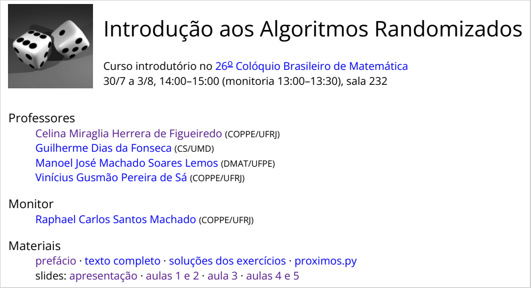
Randomized Algorithms

Computers are deterministic: set of instructions of algorithm applied to input determines its computation and output

The world we live in is full of random events that lack predictability, or a well-defined pattern

Computer scientists allow algorithms to make random choices to improve their efficiency

A randomized algorithm flips coins to compute a solution that is correct with high probability



Introdução aos Algoritmos Randomizados

Curso introdutório no 26º Colóquio Brasileiro de Matemática
30/7 a 3/8, 14:00–15:00 (monitoria 13:00–13:30), sala 232

Professores
Celina Miraglia Herrera de Figueiredo (COPPE/UFRJ)
Guilherme Dias da Fonseca (CS/UMD)
Manoel José Machado Soares Lemos (DMAT/UFPE)
Vinícius Gusmão Pereira de Sá (COPPE/UFRJ)

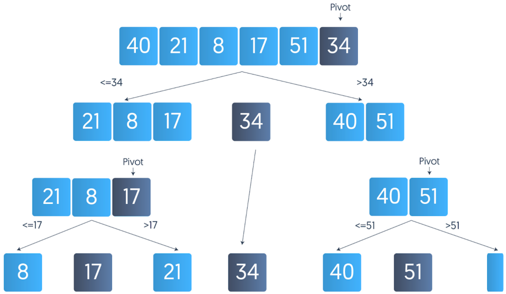
Monitor
Raphael Carlos Santos Machado (COPPE/UFRJ)

Materiais
prefácio · texto completo · soluções dos exercícios · proximos.py
slides: apresentação · aulas 1 e 2 · aula 3 · aulas 4 e 5

Brazilian Mathematics Colloquium, 2007

Sorting and Primality

Las Vegas Quicksort:
correct answer
expected time



Monte Carlo Primality Test:
expected answer
deterministic time

PSEUDOPRIME(n)

if MODULAR-EXPONENTIATION$(2, n-1, n) \not\equiv 1 \pmod{n}$
  return COMPOSITE // definitely
else return PRIME // we hope!

Cormen, Leiserson, Rivest, Stein, 2009

Trading hardness for randomness

Avi revolutionized our understanding of the role of randomness in computation

Every randomized polynomial time algorithm can be efficiently derandomized, made fully deterministic

Trade-off between hardness versus randomness:

If there exists a hard enough problem, then randomness can be simulated by efficient deterministic algorithms; conversely, efficient deterministic algorithms even for specific problems with known randomized algorithms would imply that there must exist such a hard problem

JOURNAL OF COMPUTER AND SYSTEM SCIENCES **49**, 149–167 (1994)

Hardness vs Randomness*

NOAM NISAN† AND AVI WIGDERSON‡

Institute of Computer Science,
Hebrew University of Jerusalem, Israel

Received February 27, 1989; revised September 26, 1993

We present a simple new construction of a pseudorandom bit generator. It stretches a short string of truly random bits into a long string that looks random to any algorithm from a complexity class C (e.g., P, NC, $PSPACE$, ...) using an *arbitrary* function that is hard for C. This construction reveals an *equivalence* between the problem of proving lower bounds and the problem of generating good pseudorandom sequences. Our construction has many consequences. The most direct one is that efficient deterministic simulation of randomized algorithms is possible under much weaker assumptions than previously known. The efficiency of the simulations depends on the strength of the assumptions, and may achieve $P = BPP$. We believe that our results are very strong evidence that the gap between randomized and deterministic complexity is not large. Using the known lower bounds for constant depth circuits, our construction yields an unconditionally proven pseudorandom generator for constant depth circuits. As an application of this generator we characterize the power of NP with a random oracle.

Avi Wigderson, 2023 Turing Award, Q&A with director of the IAS

I am both a mathematician and a computer scientist

I study the mathematical foundations of computing

I prove theorems to understand computation, computational processes also in nature

Could a Nobel go to innovations of computing applied to a natural science?

My three decades in this field have been a continuous joyride, with fun problems, brilliant researchers, and many students, postdocs, and collaborators who have become close friends

I'm lucky to be part of a dynamic community

watch

O Problema do Milênio sobre Intratabilidade Computacional

Celina Miraglia Herrera de Figueiredo

COPPE
UFRJ