NP-complete problem

mathematics

Written and fact-checked by The Editors of Encyclopaedia Britannica

Last Updated: Article History

Key People: Richard Karp • Stephen Arthur Cook

Related Topics: computation

See all related content →

NP-complete problem, any of a class of computational problems for which no efficient solution algorithm has been found. Many significant computer-science problems belong to this class—e.g., the traveling salesman problem, satisfiability problems, and graph-covering problems.

So-called easy, or tractable, problems can be solved by computer algorithms that run in polynomial time; i.e., for a problem of size n, the time or number of steps needed to find the solution is a polynomial function of n. Algorithms for solving hard, or intractable, problems, on the other hand, require times that are exponential functions of the problem size n. Polynomial-time algorithms are considered to be efficient, while exponential-time algorithms are considered inefficient, because the execution times of the latter grow much more rapidly as the problem size increases.

A problem is called NP (nondeterministic polynomial) if its solution can be guessed and verified in polynomial time; nondeterministic means that no particular rule is followed to make the guess. If a problem is NP and all other NP problems are polynomial-time reducible to it, the problem is NP-complete. Thus, finding an efficient algorithm for any NP-complete problem implies that an efficient algorithm can be found for all such problems, since any problem belonging to this class can be recast into any other member of the class. It is not known whether any polynomial-time algorithms will ever be found for NP-complete problems, and determining whether these problems are tractable or intractable remains one of the most important questions in theoretical computer science. When an NP-complete problem must be solved, one approach is to use a polynomial algorithm to approximate the solution; the answer thus obtained will not necessarily be optimal but will be reasonably close.

The Editors of Encyclopaedia Britannica

This article was most recently revised and updated by Erik Gregersen.

Home › Science › Mathematics

computational complexity

Actions

Written and fact-checked by The Editors of Encyclopaedia Britannica

Last Updated: Article History

Table of Contents

Listen to article 2 minutes

Key People: Manuel Blum • Leslie Valiant • Andrew Chi-Chih Yao • Juris Hartmanis • John Hopcroft

Related Topics: time complexity • P versus NP problem • size • computation

See all related content →

computational complexity, a measure of the amount of computing resources (time and space) that a particular algorithm consumes when it runs. Computer scientists use mathematical measures of complexity that allow them to predict, before writing the code, how fast an algorithm will run and how much memory it will require. Such predictions are important guides for programmers implementing and selecting algorithms for real-world applications.

Computational complexity is a continuum, in that some algorithms require linear time (that is, the time required increases directly with the number of items or nodes in the list, graph, or network being processed), whereas others require quadratic or even exponential time to complete (that is, the time required increases with the number of items squared or with the exponential of that number). At the far end of this continuum lie intractable problems—those whose solutions cannot be efficiently implemented. For those problems, computer scientists seek to find heuristic algorithms that can almost solve the problem and run in a reasonable amount of time.

Further away still are those algorithmic problems that can be stated but are not solvable; that is, one can prove that no program can be written to solve the problem. A classic example of an unsolvable algorithmic problem is the halting problem, which states that no program can be written that can predict whether or not any other program halts after a finite number of steps. The unsolvability of the halting problem has immediate practical bearing on software development. For instance, it would be frivolous to try to develop a software tool that predicts whether another program being developed has an infinite loop in it (although having such a tool would be immensely beneficial).

The Editors of Encyclopaedia Britannica

This article was most recently revised and updated by Erik Gregersen.

Richard Karp

Actions

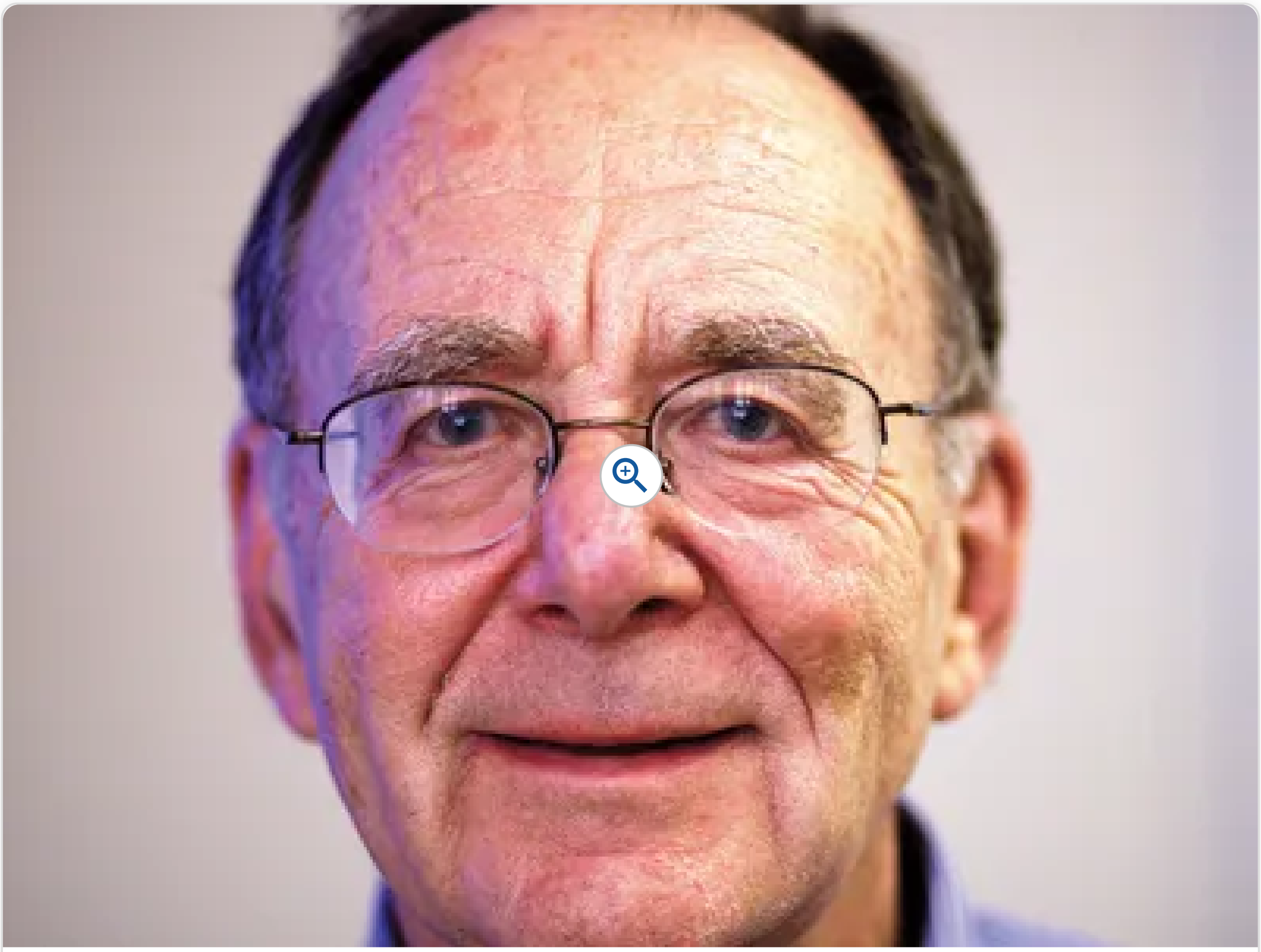
American mathematician and computer scientist

Also known as: Richard Manning Karp

Written by William L. Hosch
Fact-checked by The Editors of Encyclopaedia Britannica
Last Updated: Article History

Table of Contents

Listen to article 2 minutes


Richard Karp

See all media

Born: January 3, 1935 (age 88) • Boston • Massachusetts

Awards And Honors: National Medal of Science (1996) • Turing Award (1985)

Subjects Of Study: NP-complete problem • optimization • recursion theory

See all related content →

Richard Karp, in full **Richard Manning Karp**, (born January 3, 1935, Boston, Massachusetts, U.S.), American mathematician and computer scientist and winner of the 1985 A.M. Turing Award, the highest honour in computer science, for "his continuing contributions to the theory of algorithms including the development of efficient algorithms for network flow and other combinatorial optimization problems, the identification of polynomial-time computability with the intuitive notion of algorithmic efficiency, and, most notably, contributions to the theory of NP-completeness." His research interests have included theoretical computer science, combinatorial algorithms, discrete probability, computational biology, and Internet algorithms.

Karp earned a bachelor's degree (1955), a master's degree (1956), and a doctorate (1959), all in mathematics, from Harvard University. After finishing his studies, he worked as a mathematician at IBM (1959–68) before moving to academia. Karp held positions at the University of California, Berkeley (1968–94), the University of Washington (1995–99), and again at Berkeley (1999–), where he returned as a University Professor. In 2012 he founded the Simons Institute for the Theory of Computing at Berkeley and served as its director until 2017.

Karp's 1972 paper "Reducibility Among Combinatorial Problems" proved that many commonly studied combinatorial problems are variants of the same problem, which implies they are all probably intractable (NP-complete problems—that is, problems for which no efficient solution algorithm is known). Karp is the author of *Complexity of Computation* (1974) and holds a patent for a type of multiconnection switching network.

In addition to the Turing Award, Karp received the Fulkerson Prize in Discrete Mathematics (1979), the U.S. National Medal of Science (1996), the Harvard University Centennial Medal (1997), the Israel Institute of Technology Harvey Prize (1998), the Carnegie Mellon University Dickson Prize in Science (2008), and Japan's Kyoto Prize (2008). He was elected to the New York Academy of Sciences (1980), the U.S. National Academy of Sciences (1980), the American Academy of Arts and Sciences (1985), the Institute of Combinatorics and Its Applications (1990), the American Association for the Advancement of Science (1991), the U.S. National

Academy of Engineering (1992), the American Philosophical Society (1994), the French Academy of Sciences (2002), and the European Academy of Sciences (2004).

William L. Hosch