The Millennium Problems IV: P Versus NP

Yi Zhern

Here's a question that may intrigue you: How do you sort a shelf of books alphabetically? One approach might be to manually go through all the books, finding the first one and then repeating the process until you reach the last. Another strategy could involve grouping five books together and sorting them in batches, subsequently comparing adjacent groups, and so forth. Both methods effectively sort the books, but it raises the question: Is there an algorithm to easily solve this problem?

Sorting, in this context, is a type of algorithm. While individuals may follow an algorithm to accomplish a task successfully, computers can execute algorithms at a much faster pace than humans. However, if the algorithm is excessively complex, the computer might still require a substantial amount of time to complete it. Hence, scientists classify problems based on the time needed for an algorithm to solve them, categorizing questions into P and NP. P refers to a class of problems for which certain algorithms can solve them in polynomial time, which can be thought of as a relatively short duration. In contrast, NP, or nondeterministic polynomial time (also known as exponential time), represents a class of tasks where no algorithm exists to solve them efficiently.

P problems exhibit two key characteristics: ease of solving and ease of checking. For instance, sorting books is a P-class problem. You can devise a strategy, invest some time, and solve it. The strategy may vary in time consumption, but it typically remains polynomial. On the other hand, NP problems are challenging to solve but easy to check. Sudoku is a classic example. While it may be difficult for us to solve, once filled, verifying the solution's correctness is straightforward. Some might think that Sudoku could be P, and it merely requires more time to solve. However, this perspective changes when the complexity of Sudoku increases. Consider a puzzle featuring 144 boxes, the time needed for solving it doesn't merely increase linearly; it rises exponentially.

It's evident that P is a subset of NP since both are easily checked. However, whether P is equal to NP is the central problem that captures the attention of scientists and mathematicians. This issue was introduced by Stephen Cook in his groundbreaking paper, "The Complexity of Theorem Proving Procedures." You might wonder: Should we prove that all NP problems are in P to demonstrate equality? The answer is no that would be way too hard. Mathematicians have instead shown that there exists another class of problems called NP-complete. These problems are special as if a fast solution is found for any NP-complete problem, it can then be used to build a quick solution for any NP problem. Thus we would prove that P=NP. One such NP-complete problem is the traveling salesman problem, where a salesman aims to visit all cities once, minimizing the distance traveled.

If NP equals P, solving every intricate problem becomes feasible. Our society would witness more efficient transportation, advancements in cancer treatment by solving protein arrangements, and even games like Candy Crush would have no challenging levels. However, the safety of your savings in the bank might become questionable in such a scenario, due to reliance on hard to solve problems underpinning computer security, which would now be easy to solve. The world would look quite different if P=NP, thus it sits as the fourth millennium problem.

References

[1] Libby Daniells. *The Travelling Salesman Problem.* 2020. URL: https://www.lancaster.ac.uk/stor-i-student-sites/libby-daniells/2020/04/21/the-travelling-salesman-problem/.

[2] Quanta Magazine. *P vs. NP: The Biggest Puzzle in Computer Science.* 2024. URL: https://www.youtube.com/watch?v=pQsdygaYcE4.

[3] Up and Atom. *P vs. NP - The Biggest Unsolved Problem in Computer Science.* 2020. URL: https://www.youtube.com/watch?v=EHp4FPyajKQ.