Course › Compu... › Efficien... › P=NP

P=NP

I mentioned earlier that we do not know whether there is a practical method for solving the subset sum problem. In particular, we do not know whether the problem is in P.

In contrast, it is easy to find an efficient method for verifying whether a purported “yes”-solution to the subset sum problem is correct. For suppose someone offers us a particular subset of the initial set, and claims that its elements add up to zero. We can check whether the claim is correct by adding up the numbers in the subset (a task that can be performed in polynomial time).

The class of problems with this feature—problems for which purported “yes”-solutions can be verified in polynomial time—is often called NP.

A fascinating property of the subset sum problem is that it is not only in NP: one can show that any problem in NP can be reduced to the subset sum problem using an algorithm than can be run in polynomial time. (NP problems like this are called NP-complete”.)

This means that if there were an efficient algorithm for solving the subset sum problem, there would be an efficient algorithm for solving every problem in NP.

In other words: it would be the case that $P = NP$.

The discovery of an efficient method for solving the problems in NP would change the world. For instance, the problem of factoring a composite number of length n into its prime factors is in NP and many of the world's cryptographic techniques depend on there being no efficient method for solving this problem.

Most mathematicians believe that it is not the case that $P = NP$, but we can’t be sure until we’ve found a proof!

Discussion

Topic: Week 9 / P=NP

Hide Discussion

Add a Post

Show all posts ⌄ by recent activity ⌄

There are no posts in this topic yet.

✖