Ben Pung

Prof. Orduz

25 November 2023

Advanced Algorithms

6.1

**1. You have been working on a problem called the XYZ problem. After some time of trying to find a fast algorithm for it, you suspect that it may be NP-Complete. Your suspicions are confirmed when you discover a simple but elegant proof of the NP-completeness of the XYZ problem. (And your proof is correct.) When you announce your result, a know-it-all colleague interrupts you, saying “What you mean to say is that your algorithm is NP-Complete. I know of dozens of efficient algorithms for the XYZ problem.” Of course, your colleague is totally wrong. Explain why.**

My colleague is wrong because having a NP-complete problem doesn’t necessarily mean that every algorithm is going to be inneficient; NP-complete just means that the problem is inherently difficult to compute and that there are no known polynomial-time algorithms that solve all instances of the problem. The problem however can be reduced and have efficient problems for other instances. The colleague saying he knows many algorithms seems to contradict the NP-completeness proof you did because him saying he knows algorithms for every instance of the problem is probably not true if the problem is really NP-complete as your proof states.

**2. Professor Gooms from A&M claims that he can solve any NP-complete problem in no time by using parallelism. “Instead of alternate universes, we use alternate cpu’s.” He has several results from his 64-core processor to prove his result. Professor Gooms has missed the point (again). What’s wrong with his claim?**

While parallelism is important for solving some types or repetition problems but it wont help with NP-complete problems. Some NP-complete probably cant be parallelized. Parallelism does not change the complexity of the problem such that the NP does not go to P.

**3. Professor Gooms has given up on the multi-processor idea, but now he has a new idea. He says “The SELECT function is actually a random-selection function that always guesses right! We’ve found that by adding a few resistors and capacitors to our random number generator, we can almost always force it to guess right!” Professor Gooms has gone off the rails again. How so?**

The SELECT function is non-deterministic in this example but the professor is getting confused with adding resistors and capacitors because those are deterministic gates. So basically the professor is confusing randomness with what generates randomness. The resistors and capacitors may give a pseudo-random select state that somehow always predicts correctly but I think this is really a deterministic function that professor has built and thought it was random.