**Complexity Classes**

The first solutions people came up with for any given problem were almost always unworkably slow. Over time, other people came up with smarter solutions that were faster. However, not every problem had a faster solution, or at least, not ones that anyone was able to find.

To deal with this, people began categorizing the problems based on how fast they could be solved. For some problems, they knew that the problems could be solved in a fast manner, while for others, they knew that there just was no fast method. For a few problems though, no one managed to figure out if or if not there was a fast way to solve the problem.

This is where P and NP come in. P is basically a group of problems that we know can be solved in polynomial time, with time complexities of , , , , , , etc. Around this group, we found another group of problems that cannot be solved in polynomial time, but if we are told a solution, we can check it in polynomial time. For some problems, we later discovered that they could also be solved in polynomial time, putting them into P. For other problems, we just do not know. This made people wonder if, one day, we will discover that all the problems in NP were actually in P, i.e. P = NP, or if NP has some problems that are genuinely harder to solve that those in P. This is the P versus NP question.

If P and NP are the same, then we will start discovering solutions to some of the hardest problems on the planet, not just theoretical ones, but real ones too, like cures to cancer. On the other hand, we would also be in some trouble. All the encryption algorithms we use today are based on NP problems specifically because they are difficult to solve.

We can think of P problems as the most simple, easy to solve puzzles and NP problems like sudoku puzzles. Finding a solution takes a long time, but if we are told an answer, checking if it is correct is pretty easy. All the problems in P are also technically in NP, since one way to check an answer is to just solve the problem.

Outside of NP are problems that are difficult to even check for correctness. This is a bit like trying to figure out the best move to make in a particular state in a chess game. Even if we were told the answer, it is so difficult to check that answer for correctness that we probably will not ever have a computer that can do it.

The odd part begins when we consider whether or not P problems and NP problems are the same. At first sight, it might seem like a dumb question to even wonder whether a given solution to a sudoku problem can be checked by actually solving the puzzle ourselves. After all, it would take much longer to solve the puzzle than to check the rows and columns. However, no one has actually been able to prove that this is true. No one has been able to prove the opposite either. For all we know, there could be a way to play sudoku which would be much faster than the methods we use now.

Essentially, the question is, if there is a polynomial time method to check if an answer to a problem is correct, does that also mean that there is a polynomial time method to find the answer in the first place?

Now to answer a few questions that might arise. Firstly, why are we saying problems like sudoku are difficult? They are, but not much. What we actually mean when we say a problem is difficult is that it becomes increasingly difficult as it gets bigger. There are problems that no matter how big they get, will remain easy to solve, even if it might not be easy for a human. Multiplying huge numbers is easy work for a computer. NP problems, like sudoku, simply get increasingly difficult the larger they get. Computers can solve small sudoku problems in milliseconds, but as the grids get bigger, even super computers fail to solve them.

Second, what do P and NP stand for. P stands for Polynomial time. These are problems where the number of steps required to find a solution are functions of the size, that is . These are functions that have a time complexity of some power of . Most importantly, they are not exponential problems with time complexity like . Exponential times are much worse.

NP stands for Non-deterministic Polynomial time, which basically means that if we had a huge number of computers checking every possible solution to the problem simultaneously, we would then be able to find the solution in polynomial time.

Thirdly, almost everyone thinks that NP and P are not equal, it’s just that we have not been able to prove it yet, one way or another. Over time, people began to realize that a large number of problems being faced by humanity were all essentially the same problem. These are called NP Complete problem, because they literally encompass all the difficult parts of the entire set of NP problems. If we manage to find a polynomial time solution to a single NP Complete problem, we basically have the solution to all of them. This means sudoku is difficult for the same reason protein folding is difficult. If we find a fast way to solve a sudoku problem, we will also have found the cure to cancer. Sadly, the fact that no one has been able to find a single fast solution to any of the NP Complete problems, which later turned out to be the same problem, hints that there just is no fast solution.

Why exactly is it difficult to prove P versus NP one way or the other? Well, proving things, is an NP problem itself, a Co-NP problem to be specific. So, it is rather difficult to do.

Beyond NP Complete, there is another group of problems that are even more difficult to prove called EXP problems, which take exponential time to even check. Chess is an EXP problem. Any problems that are at least as hard as NP complete are called NP Hard problems.

There is another class where, instead of checking for right answers, it is easy to prove wrong answers. These are called Co-NP problems and they may or may not be the same thing as NP problems. Again, we do not know. There is also a subgroup of Co-NP problems that are Co-NP Complete problems.

Then we have PSPACE problems, which are ones that can be solved if we have an unlimited amount of time, but have a polynomial space complexity, and BPP problems, which can be solved probabilistically in polynomial time. BPP may or may not be the same as P.

In between all of this there are many other layers of problems of different difficulties. There are also problems that are layers and layers more difficult than EXP problems, even some that are just not solvable given any amount of time or space.