P and NP problems

In computer science, there exist some problems whose solutions are not yet found, the problems are divided into classes known as **Complexity Classes**. In complexity theory, a Complexity Class is a set of problems with related complexity. These classes help scientists to group problems based on how much time and space they require to solve problems and verify the solutions. It is the branch of the theory of computation that deals with the resources required to solve a problem. The P and NP are types of complexity classes.

1.P Class :

The P in the P class stands for **Polynomial Time.** It is the collection of decision problems (problems with a "yes" or "no" answer) that can be solved by a deterministic machine in polynomial time.
these problems can be solved in time $O(n^k)$ in worst-case, where **k** is constant.

Features:

1. The solution to P problems is easy to find.
2. P is often a class of computational problems that are solvable and tractable. Tractable means that the problems can be solved in theory as well as in practice. But the problems that can be solved in theory but not in practice are known as intractable.

This class contains many natural problems:

1. Calculating the greatest common divisor.
2. Finding a maximum matching.
3. Decision versions of linear programming.

2.NP class:

The NP in NP class stands for **Non-deterministic Polynomial Time**. It is the collection of decision problems that can be solved by a non-deterministic machine in polynomial time.

Features:

1. The solutions of the NP class are hard to find since they are being solved by a non-deterministic machine but the solutions are easy to verify.
2. Problems of NP can be verified by a Turing machine in polynomial time.

This class contains many problems that one would like to be able to solve effectively:

1. **Boolean Satisfiability Problem (SAT).**
2. **Hamiltonian Path Problem.**
3. **Graph coloring.**

Use case : Traveling salesman problem

The traveling salesman problem (TSP) is a well-known optimization problem in computer science and operations research. It involves finding the shortest possible route that visits a set of cities exactly once and returns to the starting city. The problem is often stated as follows:
Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the starting city.

no efficient algorithm has been found that solves TSP in polynomial time for all possible inputs. Therefore, TSP is believed to be a problem that cannot be solved in polynomial time so its classified as NP problem

P versus NP:

Every decision problem that is solvable by a deterministic polynomial time algorithm is also solvable by a polynomial time non-deterministic algorithm.

The problem belongs to class **P** if it's easy to find a solution for the problem. The problem belongs to **NP**, if it's easy to check a solution that may have been very tedious to find.

P and NP are related because every problem in P is also in NP, but it is not known whether every problem in NP is in P. If a problem in NP can be shown to be in P, it is said to be "solved in polynomial time" and is considered to be a P problem. However, if no polynomial time algorithm can be found for an NP problem, it is said to be an NP-complete problem. NP-complete problems are considered to be some of the hardest problems in computer science, and finding an efficient algorithm for solving them would have significant practical implications in fields such as cryptography and optimization.

Briefly, The complexity class P (polynomial time) consists of problems that can be solved in polynomial time, which means that the time required to solve the problem grows no faster than a polynomial function of the size of the input O(n^k) . In other words, if the size of the input is doubled, the time required to solve the problem increases by at most a factor of some polynomial.

On the other hand, the complexity class NP (nondeterministic polynomial time) consists of problems that can be verified in polynomial time, which means that given a solution to the problem, it can be checked in polynomial time whether the solution is correct or not.