

## Basic Concepts

There are two groups in which a problem can be classified. The first group consists of the problems that can be solved in polynomial time. For example : searching of an element from the list  $O(\log n)$ , sorting of elements  $O(\log n)$ .

The second group consists of problems that can be solved in non-deterministic polynomial time. For example : Knapsack problem  $O(2^{n/2})$  and Travelling Salesperson problem ( $O(n^2 2^n)$ ).

- Any problem for which answer is either yes or no is called decision problem. The algorithm for decision problem is called **decision algorithm**.
- Any problem that involves the identification of optimal cost (minimum or maximum) is called optimization problem. The algorithm for optimization problem is called **optimization algorithm**.
- Definition of P** - Problems that can be solved in polynomial time. ("P" stands for polynomial). The polynomial time is nothing but the time expressed in terms of polynomial.

Examples - Searching of key element, Sorting of elements, All pair shortest path.

- Definition of NP** - It stands for "non-deterministic polynomial time". Note that NP does not stand for "non-polynomial".

Examples - Travelling Salesperson problem, Graph coloring problem, Knapsack problem, Hamiltonian circuit problems.

- The NP class problems can be further categorized into NP-complete and NP hard problems.
- A problem D is called NP-complete if -
  - It belongs to class NP
  - Every problem in NP can also be solved in polynomial time.
- If an NP-hard problem can be solved in polynomial time then all NP-complete problems can also be solved in polynomial time.
- All NP-complete problems are NP-hard but all NP-hard problems cannot be NP-complete.

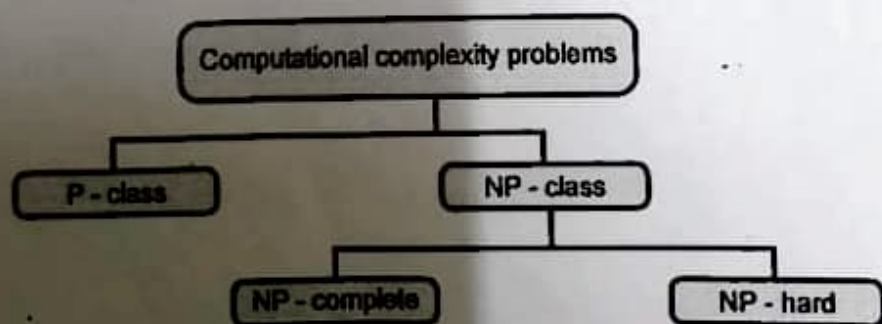


Fig. 7.1.1

- The NP class problems are the decision problems that can be solved by non-deterministic polynomial algorithms.
- **Computational Complexity** - The computational problems is an infinite collection of instances with a solution for every instance.

In computational complexity the solution to the problem can be "yes" or "no" type. Such type of problems are called **decision problem**. The computational problems can be **function problem**. The function problem is a computational problem where single output is expected for every input. The output of such type of problems is more complex than the decision problem.

The computational problems also consists of a class of problems, whose output can be obtained in polynomial time.

- **Complexity classes** - The complexity classes is a set of problems of related complexity. It includes function problems, P classes, NP classes, optimization problem.
- **Intractability** - Problems that can be solved by taking a long time for their solutions is known as intractable problems.
- If NP is not same as P then NP complete problems are called intractable problems.



## Difference between P and NP Class Problems

| P Class Problems ✓  | NP Class Problems   |
|---|---|
| An algorithm in which for given input the definite output gets generated is called Polynomial time algorithm(P class) | An algorithm is called non deterministically polynomial time algorithm (NP class) when for given input there are more than one paths that the algorithm can follow. Due to which one can not determine which path is to be followed after particular stage. |
| All the P class problems are basically deterministic.   | All the NP class problems are basically non deterministic.  |
| Every problem which is a P class is also in NP class.   | Every problem which is in NP is not the P class problem.  |
| P class problems can be solved efficiently.   | NP class problems can not be solved efficiently as efficiently as P class problems.   |
| Examples : Binary Search, Bubble Sort   | Examples : Knapsack problem, Traveling Salesperson problem.   |