NP-Completeness

9/6/21

Polynomial-time algorithms:

On input size of n, their worst-case running time is $O(n^k)$, for some constant k.

e.g. Bubble Sort:

$k=2$ $O(n^2)$

Linear search: $k=1$ $O(n)$

Matrix chain Multiplication:

$k=3$ $O(n^3)$

Most of the algorithms have this kind of polynomial time which we use in solving the problems.

But can we solve all problems in polynomial time?.

The answer is no.

For example, there are problems, such as Turing's famous "Halting Problem" that cannot be solved by any computer, no matter how much time we allow.

NP - Completeness

9/6/21

Turing's Halting Problem:

Given a program/algorithm will ever halt or not?

Halting means that the program on certain input will accept it and halt or reject it and halt and it would never go into an infinite loop.

Basically halting means terminating.

So can we have an algorithm that will tell that the given program will halt or not in terms of turing machine, will it terminate when run on some machine with some particular given input string.

The answer is no. We cannot design a generalized algorithm which can appropriately say that given a program will ever halt or not?

NP-Completeness

9/6/21

These are also problems that can be solved, but not in time $O(n^k)$ for any constant k.

Generally, we think of problems that are solvable by polynomial time algorithms as being tractable, (easy) and problems that require superpolynomial time as being intractable, or hard.

NP-complete: The class of problems called NP-complete problems whose status is unknown.

No polynomial-time algorithm has yet been discovered for an NP-complete problem, nor has anyone yet been able to prove that no polynomial time algorithm can exist for any one of them.

NP-Complete

9/6/21

In following pair of problems one is solvable in polynomial time and other is NP-Complete.

(1.) Shorstest vs longest simple paths with even negative weights, we can find shortest paths from a single source in a directed graph $G = (V.E)$ in $O(V.E)$ time.

Finding a longest simple path between two vertices is difficult, however merely determining whether a graph contains a simple path with at least a given number of edges is NP-Complete

(2.) Euler toure vs hamiltonian cycle: An Euler toure of a connected, directed graph $G = (V, E)$ is a cycle that traverses each edge of G exactly once, although it is allowed to visit each vertex more than once.

NP- Complete

9/6/21

we can determine whether a graph has an Euler tour in only $O(E)$ time., and in fact we can find edges of the Euler tour in $O(E)$ time.

A hamiltonian cycle of a directed graph $G = (V, E)$ is a simple cycle that contains each vertex in V. Determining whether a directed graph has a hamiltonian cycle is NP-complete

P class: The class P consists of those problems that are solvable in polynomial time. Solved in time $O(n^k)$ for some constant k.

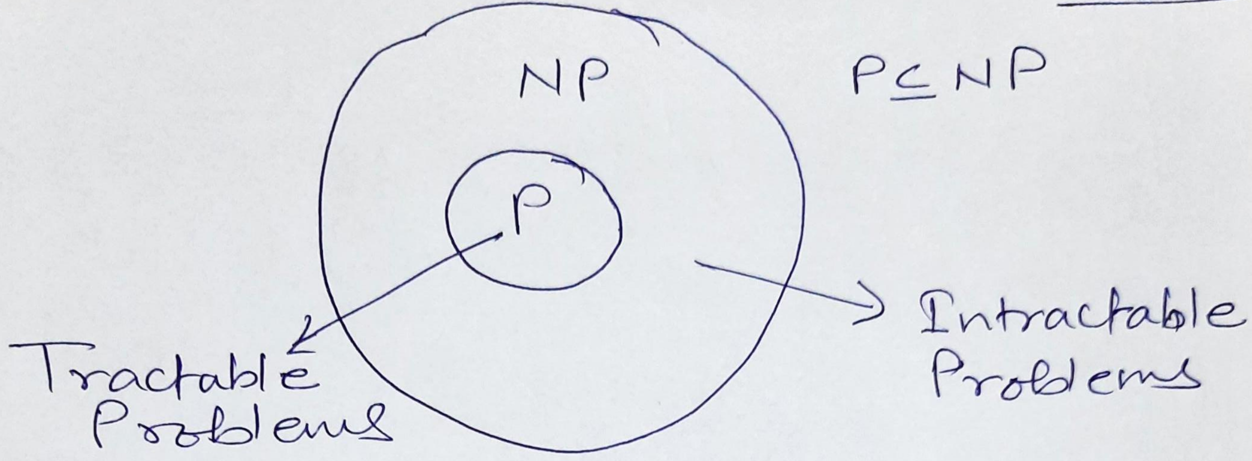
NP class: The class NP consists of problems that are "verifiable" in polynomial time.

NP Complete: NP- Hard problems

9/6/21

NP-Complete

$P \subseteq NP$



Algorithms

- Polynomial time
  - linear search $O(n)$
  - Binary Search $O(\log n)$
  - selection Sort $O(n^2)$
  - Heap Sort $O(n \log n)$
  - Matrix Chain Mult $- O(n^3)$
- Non Polynomial time or Exponential time
  - 0/1 Knapsack $- 2^n$
  - TSP $- 2^n$
  - Graph Coloring $- 2^n$
  - Sum of Subsets $- 2^n$
  - Hamiltonian cycle $- 2^n$