Final Assighment CSE 373.9 Spring 2020 1711661
Date :

Q8

P(polynomial) problem → P problems refer to problems where an algorithm would take a polynomial amount of time to solve. or Big-O is a polynomial like O(1), O(n), O(n^2) etc.

NP(Non-deterministic polynomial) problems → Non-deterministic polynomial time solving, problem which can't be solved in polynomial time. But NP problems are checkable in polynomial time means that given a solution of a problem, we can check that wheather the solution is correct or not in polynomial time.

NP-hard is the class of decision problems to which all problems in NP can be reduce to in polynomial time by a deterministic Turing machine

NP-complete is the class of decision problems in NP to which all other problems reduce to in polynomial time by a deterministic turing machine.

A language B is NP-complete if it satisfy two conditions

- B is in NP
- Every A in NP is polynomial time reducible to B

If a language satisfies the second property, but not necessarily the first one, the lanuage B is known as NP-Hard. A search problem B is NP-hard if there exists some NP-complete problem A that Turing reduce to B

Below is a venn diagram of different classes

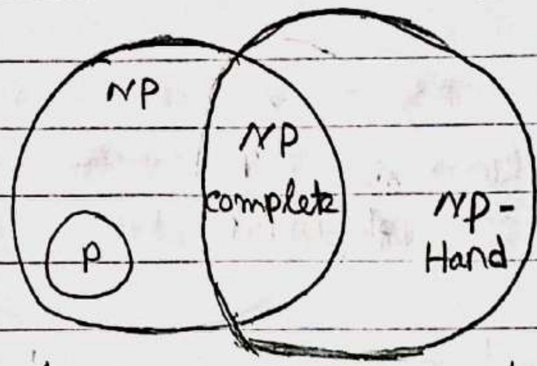

fig → venn diagram of P, P-hand, NP, Np-complete.

Some examples of Np-complete and Np-Hand problems

problems like - Knapsack problem, Hamiltonian path problem. Travelling salesman problem, The circuit-satietiability problem, Boolean satisfy problem. etc.

The difference between Np-Hand and Np-complete is

Np-Hand	Np-Complete
(I) Np-Hard problems can be solved if and only if there is a Np-Complete problem can be reduce into in polynomial time.	(I) A problem is Np-Complete if any problem in Np can be reduce to it in polynomial time and its a Np.
(II) To solve, Np-Hard, it must be a Np-problems.	(II) To solve, Np-Complete, it must be both Np and Np-complete.
(III) Np-problems are → Halting problems, Traveling salesman	(III) Np-completes problems are → KnapSack problem, Hamiltonian path problem etc.