Date : / /
Page No.

Unit ⇒ (5) (Important Topics)

Computation

★ 1st Priority Topics :→ P, NP, NP-hard, NP-Complete Problem

★ 2nd Priority Topic → Approximation Algorithms

★ P, NP, NP-Hard, NP-Complete Problem:

→ In Computer Science, there are exists some problems whose solutions are not yet found, the problems are divided into Complexity Classes.

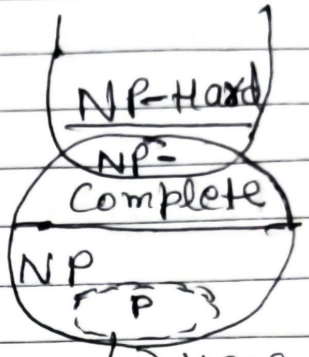
→ In Complexity theory, a Complexity class is a set of problems with related Complexity.

→ The common resources are time & space, meaning how much time the algorithm takes to solve a problem and the memory usage.

→ The time Complexity is used to describe the number of steps required to solve a problem, but it can also be used to describe, how long it

-> The space complexity of a problem or algorithm described how much memory is required for the algorithm to operate.

Complexity Classes



-> Types of Complexity Classes:

1) P Class.
2) NP Class.
3) NP-Hard.
4) NP- Complete.

1) => P-Class :-> The (P) stands for Polynomial time.

-> It is the collection of decision problems (problems with Yes or no) that can be solved by a determinid machine in a polynomial time.

-> Features:

-> It is easy to find the solution by this problem.

-> It is easily solvable & Tractable (Tractable means, the problems that can be solved using in theory

or in practical).

2) NP Class :→ It stands for Non-Deterministic polynomial-time.

→ It is the collection of decision problems that can be solved using non-deterministic machine in polynomial-time.

3) NP-Hard Class:

→ All NP-hard problems are not in (NP).
→ It taked long-time to check them, It means, if a solution for an NP-hard problem is given, then it taked a long time to check whether it is right or not.
→ A problem (A) is in NP-hard if, for every problem (L) in NP, there exists a polynomial-time reduction from (L) to (A).

⇒ Some of the Common problems in NP-Hard are:

1) Halting Problem.
2) Qualified Boolean formula.
3) No Hamilton cycle.

4) NP- Complete Cladd:

→ It id complete, if both (NP) and NP-Hard, NP problemd are the hard problemd in (NP).

→ featured:

→ NP- Complete problemd are special ad any problem in (NP) cladd can be trandformed or reduced into (NP)- Complete problemd in polynomial-time.

→ It id time conduming.

→ dome Common problem are:

(1) Hamilton Cycle.
(2) datidfiability.
(3) Vertex Cover.

Complexity Cladd	feature/characterdtic
P	Eadily dalvable
NP	Yed, andwerd can be check in polynomial-time.
NP- Hard	All NP- Hard problem are not in NP & it taked a long time to check.
NP- Complete	A problem that id (NP) or (NP-Hard) id NP- Complete.

2nd Priority Topic:

★ Approximation Algorithms:

→ It is a way of dealing with NP-Complete for an optimization problems. nearly

→ Featured:

1) It guaranteed to run in polynomial-time through it does not guaranteed the most effective solution.

2) It guaranteed to seek out high accuracy & top quality solutions (within 1% of optium).

3) It is used to get an answer near the (optimal) solution of an optimization problem in polynomial-time.

⇒ Performance Ratio for Approximation Algorithm:

→ Scenario (1):

1) Let we are working on an optimization problem in which each potential soln had a cost, and wish to find a near optimal solution.

2) The algorithm for a problem had an

appropriate ratio of P(n) if, for any input size (n), the cost (C) for any solution produced by the algorithm is within a factor of P(n) of the cost C* of an optimal solution as follows:

$$\max(C/C^*, C^*/C) <= P(n)$$

→ Scenario(2):

→ If an algorithm reached an approximate ratio of P(n), then we call it on a P(n) - approximation algorithm.

↳ for a maximization problem, $0 < C < C^*$, & the ratio of C^*/C gives the factor by which the cost of an optimal solution is larger than the cost of the approximate algorithm.

↳ for a minimization problem, $0 < C^* < C$, and the ratio of C/C^* gives the factor by which the cost of an approximate solution is largest than the cost of optimal solution.

Date : / /
Page No.

→ Some examples of Approximation Algorithm:

1) The vertex cover Problem.
2) Travelling Salesman Problem.
3) The set-Covering Problem.
4) The Subset-Sum problem.

FOLLOW INFINITY TUTIONS