Page No:
Date:

Tutorial No: 10

Q.1. How p-class problems are different from NP-class.

→ (1) Every decision problem that is solvable by a deterministic polynomial time algo. is also solvable by a polynomial time non-deterministic algo.

(2) All problems in P can be solved with polynomial time algorithms, whereas all problems in NP-P are intractable.

(3) It is not known whether P=NP. However, many problem are known in NP with the property that if they belong to P, then it can be proved that P=NP.

(4) If $P \neq NP$, there are problems in NP that are neither in P nor in NP-complete.

(5). The problem belongs to class P if its easy to find a sol^n for the problem. The problem belongs to NP, if its easy to check a sol^n that may have been very tedious to find.

Q2 What are P & NP-class problems & explain their concept with suitable examples.

→ (1) P-class:

The class P consists of those problems that are solvable in polynomial time i.e. these problems can be solved in time $O(n^k)$ in worst-case, where k is constant.

These problems are called tractable while others are called intractable or superpolynomial.

Formally, an algo. is polynomial time algo., if there exist a polynomial $p(n)$ such that the algo. can solve any instance of size n in a time $O(p(n))$.

Problem requiring $\Omega(n^{50})$ time to solve are essentially intractable for large n. Most known polynomial time algo.

run in time $O(n^k)$ for fairly low value of k.
Example:- stable roommate problem, its polynomial-time to match without a tie, but not when ties are allowed. Or when we include roommate preferences like married couples. Still another factor to consider is the size of k relative to n. If the I/p size is going to be near k, the algo. is going to behave more like an exponential.

② NP-class:
The class NP consists of these problems that are verifiable in polynomial time. NP is the class of decision problems for which it is easy to check the correctness of a claimed answer, with the aid of a little extra information. Hence, we aren't asking for a way to find a solⁿ, but only to verify that an alleged solⁿ really correct.

Every problem in this class can be solved in exponential time using exhaustive search.
Example: Integer factorization & Graph Isomorphism are examples of NP-class. Both of these have two important characteristics: complexity is $O(k^n)$ for some k & their results can be verified in polynomial time, those two facts place them in NP-class.

Q.3 Show that the travelling salesman problem is an NP-complete problem.

→ The travelling salesman problem consists of a salesman & set of cities. The salesman has to visit each one of cities starting from a certain one & returning to the same city. The challenges of problem is that the travelling salesman

Page No.:
Date:

wants to minimize the total length of trip.

Proof: To prove TVSP is NP-complete, first we have to prove that TSP belongs to NP. In TSP, we find a toue & check that the toue contains each vertex once. Then the total cost of edges of the toue is minimum. finally, we check if the cost is minimum. This can be completed in polynomial time. Thus TSP belongs to NP.

Secondly we have to prove TSP is NP-hard. To prove this, one way is to show Hamiltonian cycle $\leq_p$ TSP. Assume $G=(V,E)$ to be an instance of Hamiltonian cycle. Hence, an instance of TSP is constructed. We create the comple graph $G'=(V,E')$, where

$$E' = \{(i,j) : i,j \in V \ \& \ i \neq j\}.$$

Thus, the cost funct^n is defined as follows –

$$t(i,j) = \begin{cases} 0, & \text{if } (i,j) \in E \\ 1 & \text{otherwise} \end{cases}$$

Now, suppose that Hamiltonian cycle exists in G. It is clear that the cost of each edge in h+ is 0 in G' as each edge belongs to E Therefore, h, has a cost of 0 in G'. Thus, if graph G has a Hamiltonian cycle; then graph G' has a foue of 0 cost.

Conversely, we assume that G' has 0 toue 'h' of cost atmost 0. The cost of edges in E' are 0 & 1 by def^n. Hence each E' are 0 & 1 def^n. Hence each edge must have a cost of 0 as the cost of h' is 0. We therefore conclude that h' contains only edges in E

We have thus proven that G has a Hamiltonian cycle if and only if G' has a toue of cost atmost 0. TSP is NP-complete.

Q.4. Explain the relationship bet^n class P, NP, NP-complete & NP-hard problem with example.

→ ① Class P: If a problem can be solved by a deterministic Turing mechanism in polynomial time, the problem belongs to the complexity class P. All problem in this class have a sol^n whose time requirement is a polynom on the i/p size n i.e. $f(n)$ is of form $a_k n^k + a_{k-1} n^{k-1} + .. a_2 n^2 + a_1 n^1 + a_0$ where $a_k, \ldots a_0$ are constant factors. The order of polynomial is the largest exponent k such that $a_k = 0$

② Class - NP: It contains all computational problems such that the corresponding decision problem can be solved in a polynomial time by a non-deterministic Turing machine.

③ NP-complete problems: These are special problems in class NP i.e they are subset of class NP. An problem p is NP-complete if:

(i) $P \in NP$ (we can solve it in polymial time by a non-deterministic Turing machine) &

(ii) All other problems in class NP can be reduced to problem P in polynomial time.

④ NP - hard problems: These are partly similar but more difficult problems than NP complete problems. They don't themselves belong to NP, but all problems in class NP can be reduced to them. Very often, the NP-hard problems really require exponential time or even worse.

Page No.:
Date:

Q.5- Explain circuit satisfiability problem.

→ Circuit SAT:

•According to given decision-based NP problem, you can design the ckt. to verify a given mentioned o/p also within the P time.

SAT (satisfiability):-

A Boolean func^n is said to be SAT if the o/p for the given value of i/p is true/high/1. F = X+YZ.

These points you have to be performed:-

① Concepts: A Boolean f^n is said to be SAT if the o/p for the given value of i/p is high/1.

② Circuit SAT ≤ p SAT: In this conversion, you have to convert circuit SAT into SAT within the polynomial time

③ SAT ≤ p circuit SAT: For the sake of verification of an o/p you have to convert SAT into CIRCUIT SAT within the polynomial time & through the circuit SAT you can get the verification of an o/p successfully.

④ SAT ∈ NPC: As you know very well, you can get the SAT through CIRCUIT SAT that comes from NP.

Reduction has been successfully made within the polynomial time from CIRCUIT SAT TO SAT. O/P has also been verified within the polynomial time as you did in above. So concluded that SAT ∈ NPC.

Q.6- Explain Clique decision problem.

→ In an undirected graph, a clique is a complete sub-graph of given graph. Complete sub-graph is connected to all other vertices.

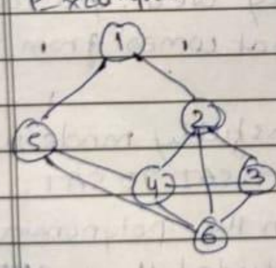
The max-clique problem is the computational problem of finding maximum clique of the graph. Max clique is used in many real-world problems.

let us consider a social networking applⁿ, where vertices represent people's profile & the edges present mutual acquaintance in a graph.

To find a max clique, one can systematically inspect all subsets, but this sort of brute-force search is too time-consuming for networks comprising more than few dozen vertices.

Analysis: Max-clique problem is a NP algo. In this 1st we try to determine a set of k distinct vertices & then we try to test whether these vertices form a complete graph. There is no polynomial time deterministic algo. to solve this problem. This is NP-complete.

Example:



Here, the sub-graph containing vertices 2,3,4 & 6 forms a complete graph. Hence, this sub-graph is clique. At this is maximum complete sub-graph of the provided graph, it's a 4-clique.