11/01/2022
Tuesday

MODULE :

Class Polynomial And Non-Polynomial {Class P & NP}

NP Hard and NP Complete

Some computational pblms are hard and difficult, for these pblms no efficient algorithms exists. A concept known as NP completeness deals with finding of an efficient algorithm for certain problems.
The notion of efficient is used to describe that a pblms algorithm's running time is proportional to the polynomial fn of it input size n. Thus for some constant $k>0$, the algorithm is efficient, if it runs in $O(n^k)$ times for input of size n

Class P

Class which contains all pblms that are solvable in polynomial time. P is the set of decision pblm with yes/no answer, i.e polynomial bound.

An algorithm is said to be polynomial bounded, if its worst case complexity is bound by a polynomial fn P of input size n in that case for each input of size n

the algorithm terminates after atmost $P(n)$ steps.
For example
$$n^2 + 13n + 10$$

V.Impt

Decision Problem

The pblms under this class have a single bit output which shows 0 or 1 i.e. the answer for the pblm is either 0 or 1.
For example some decision pblms are:
i) Given 2 sets of strings S_1 and S_2, thus S_2 is a substring of S_1
ii) Given 2 sets of elements S_1 and S_2, thus both the sets contain same no. of elements

Any problem that involves the identification of an optimal value of a given cost fn is known as optimization pblm.
For example:
Given a weighted graph G, and an integer i. Thus G have a minimal spanning tree of weight of atmost i.

Basic Concepts

Algorithms are divided into 2 gps based on their computing time. The first gp consist of pblms whose soln times are

bounded by polynomials of small degree
Eg: Order searching - O(log n)
Polynomial Evaluation - O(n) etc..

The 2nd gp is made up of pblms whose best known algorithms are ~~known po~~ Non-Polynomial.
Eg: Travelling Salesperson pblm - $O(n^2 2^n)$
Knapsack Pblm - $O(2^{n/2})$

V.Impt
There are 2 classes of pblms: np hard and NP complete.

The pblm is NP complete, has the ppty that it can be solved in polynomial time iff all other NP complete pblms can also be solved in ~~AND~~ polynomial time.

If an NP hard pblm can be solved in polynomial time. Then all NP complete pblms can be solved in polynomial time. All NP complete pblms are NP hard, but some NP hard pblms are not known to be NP complete

19/01/2022
Wednesday

Deterministic / Non-Deterministic

Non-Deterministic Algorithm

Deterministic Algorithms are algorithm which have the ppty that the result of every operation is uniquely defined.

Non-deterministic algorithms are algorithm which contains operations whose outcomes are not-uniquely defined but are limited to specified set of possibilities.

To specify such algorithms 3 new functions are introduced:

i) Choise of S
Arbitrarily chooses one of the elements of set S.

ii) Failure
Signals the unsuccessful completion

iii) Success
Signals a successful completion

Whenever there is a set of choices that leads to a successful completion, then one such set of choices is always made and the algorithm terminates successfully. A non-deterministic algorithm terminates unsuccessfully iff ∄ no set of choices leading to a success signal

Algorithm Non-Deterministic-Search

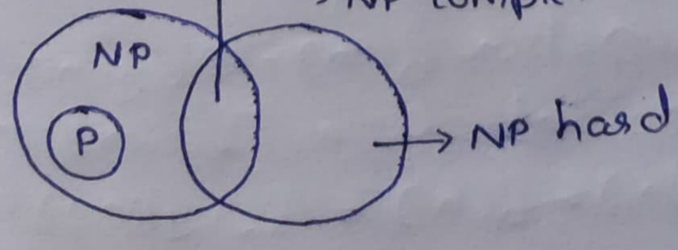
```
1 j = choice (1,n)
2 If a(j) = S
    {
      write (j)
      success;
    }
3 write (0) ; failure
```

The classes NP hard & NP complete

P is the set of all decisions solvable by deterministic algorithms in polynomial time. NP is the set of all decision pblms solvable by non-deterministic algorithms in polynomial time. ie $P \subseteq NP$.

Relationship b/w P and NP

Relationship b/w P, NP, NP complete & NP hard



There are NP hard pblms, that are not NP complete. Only a decision pblm can be NP complete. An optimization pblm may be NP hard.

Maximum Clique Problem

A maximal complete subgraph of a graph $G = (V, E)$ is a clique, the size of the clique is the no. of vertices in it. The max clique pblm is an optimization pblm, that has to determine the size of largest clique in it. The corresponding decision pblm is to determine whether G has a click of size of atleast k for some given k.