**NATIONAL INSTITUTE OF TECHNOLOGY, DELHI**



**ASSIGNMENT – 5 : DESIGN AND ANALYSIS OF ALGORITHMS**

**P and NP problems**

Submitted by : - dimpal kataniya

Roll no. : - 181210022

Cse 2nd year(group-1)

Undecidable problem

Tractable problem

Intractable problem

Decidable problem

problem

terminology

(algorithm exist) (algorithm does not exist)

If there exist at least If a problem is not tractable

one polynomial bound then it is intractable

algorithm O(n^k). O(n^logn) or O(c^n).

**optimization and decision problems:-**

* Every optimization problem can be converted into decision problem.
* Example :-
* For travelling salesman problem :-
* Optimization problem :-

For a graph G shortest path covering all vertices exactly once.

* Decision problem :-

Is there any shortest path covering all the vertices exactly once of length at most k.

* We convert every optimization problem to decision problem then we find if there exist a polynomial time solution for decision problem or not. If there is not any easy solution for decision problem there can not be an easy solution for optimization problem.
* If optimization problem is easy then decision problem will also be easy.
* If decision problem is hard optimization problem will surely be hard.

**Verification algorithm:-**

In future if any one find the polynomial time solution of an intractable problem then we have to check if the answer computed by this polynomial time algorithm is correct or not, to solve this decision problem we will use a verification algorithm.

This verification algorithm should be a polynomial time algorithm so that we can find the solution of our decision problem in polynomial time.

**P and NP introduction :-**

**P :-**

set of all decision problems for which there exist a polynomial time algorithm to solve them.

**Examples :-**

1. Minimum spanning tree problems using prim’s and Kruskal algorithms.
2. Fractional knapsack problem.

**NP :-**

set of all decision problems for which there exist a polynomial time verification algorithm.

**Examples :-**

1. If we are able to solve a problem in polynomial time, we will surely be able to verify in polynomial time, so every P problem will also be a NP problem.
2. Travelling salesman problem :- we are not able to find polynomial time solution for this problem but we can verify this in polynomial time so this a NP problem but not a P problem.

(NP-P) :- as of now these problems are intractable.

Till now we believe P c NP

P = NP or not is still an open question.

**Polynomial time reduction algorithm :-**

To prove P = NP we have to prove that every problem which lies in NP can be solved in polynomial time

There are millions of NP problems we can not solve each problem to prove this, here comes reduction.

A problem ‘A’ is said to be polynomial time reducible to a problem ‘B’ if :-

1. Every instance ‘a’ of ‘A’ can be transformed to some instance ‘b’ of ‘B’ in polynomial time.
2. Answer of ‘a’ is ‘YES’ if and only if answer of ‘b’ is ‘YES’.

So

If A is reduced to B in polynomial time then :-

* If B is easy then A is also easy.
* If B is in P then A is also in P.
* If this is proven that A can not be solved in polynomial time then B is also can not be solved in polynomial time.
* If A is not in P then B is also not in P.

**NP hard and NP complete problems :-**

**NP hard :-**

If every problem in NP can be polynomial time reducible to a problem ‘A’ then ‘A’ is called a NP hard problem.

* If ‘A’ could be solved in polynomial time then every problem in NP can be solved in polynomial time so

P = NP

**NP complete :-**

If a problem lies in NP and also a NP hard problem then this problem is called NP complete problem.

**Some important points :-**

1. If NP hard or NP complete problem is solved in polynomial time then NP = P.
2. If NP or NP complete problem is proven to be not solvable in polynomial time then P NP.
3. If ‘A’ is a NP hard problem and ‘A’ can be reduced in polynomial time in ‘B’ then be is also called a NP hard problem. If this ‘B’ is a NP problem then ‘B’ will be NP complete problem.

NP

NP complete

NP hard

**Some well known NP complete problems :-**

CIRCUIT - SAT

SAT

3 CNF - SAT

SUBSET SUM

CLIQUE

VERTEX COVER

HAMILTONIAN CYCLE

TSP