**THE CLASSES P AND NP**

The Problems solvable in polynomial time on a typical computer are exactly the same as the problems solvable in polynomial time on a Turing Machine. The problems which cannot be solvable in polynomial time are called ‘intractable problems’.

**Problems Solvable in Polynomial time:**

The time complexity or running time T(n) of a Turing machine is after taking the input w of length n, M halts after making atmost T(n) moves, regardless of whether or not M accepts.

**For eg:** The complexity of bubble sort algorithm is T(n)=n2

A language is in class P if there is some polynomial T(n) such that L=L(M) for some deterministic TM M of time complexity T(n).

Example: Kruskal’s Algorithm

The problems with efficient solutions are comes under the class P. Consider one such problem MWST (Minimum Weight Spanning Tree).

Consider a graph with n nodes and e edges. Each edge has a weight represented in integers. A spanning tree is subset of edges such that all nodes are connected through edges where there are no cycles. A Minimum weight spanning tree has the least possible total edge weight of all spanning trees.

A well known greedy algorithm is there called Kruskal’s Algorithm for finding a MWST.

**Algorithm:**

1. Maintain the ‘connected component’ for each node . Initially every node is a connected component by itself.
2. Sort the edges in ascending order of their weights.
3. Select the lowest-weight edge that has not yet been visited.
   1. Select that edge for the spanning tree.
   2. Merge the two connected components involved.
4. Repeat the step 3 until all the edge have been considered by which the number of edges selected is one less than the number of nodes.

**Example:**Consider the graph given below. Find MWST.

12

10 18 20

13

15

**Solution:**

The ascending order of their weights are,

W= {10,12,13,15,18,20}

Number of nodes N=4

Select the edge (1,3) with weight 10

10

2. Select the second lowest weight from the list and check whether it forms any cycle or already included in the list. If it is so, include that edge to the forest(Group of edges).

12

10

3. Repeat the step (2) and (3) till all the nodes are visited.

12

10

13

**Non Deterministic Polynomial Time:**

A language L is in class NP(nondeterministic polynomial) if there is a nondeterministic TM M and a polynomial time complexity T(n) such that L=L(M).But it is true that P C NP

But it appears that NP contains many problems not in P. The reason is that a NTM running in polynomial time has ability to guess an exponential number of possible solution to a problem and check each one in polynomial time.

**Example: Travelling Salesman problem.**

Given a graph G=(V,E) where V-number of vertices and E-the edge connecting two vertices(v1,v2). A salesperson has to visit each vertex exactly once and returns to its starting point with minimum distance.

This can be implemented by a TM with question that “whether the graph has a Hamilton circuit of total weight at most w? A Hamilton circuit is a set of edges that the nodes into a single cycle ,with each node appearing once.

12

15

10 20

18

18

In this graph ,we have only 1 hamilton circuit(1,2,4,3,1)

Where the weight is given by W=15+20+18+10=63

Thus If w≥63,the answer is Yes

If w < 63 the answer is No.

For a M-node graph, the number of distinct cycle grows as 0(M1). For a nondeterministic computer, it is possible to guess a permutation of the nodes and compute the total weight for the cycle of nodes in that order.So using mulititape NTM.It is possible to guess a permutation in 0(n2) steps and checks its total weight in the same time. Thus TSP comes under the class NP