Ch ⇒ 4 (Important Topics)

Dynamic Programming

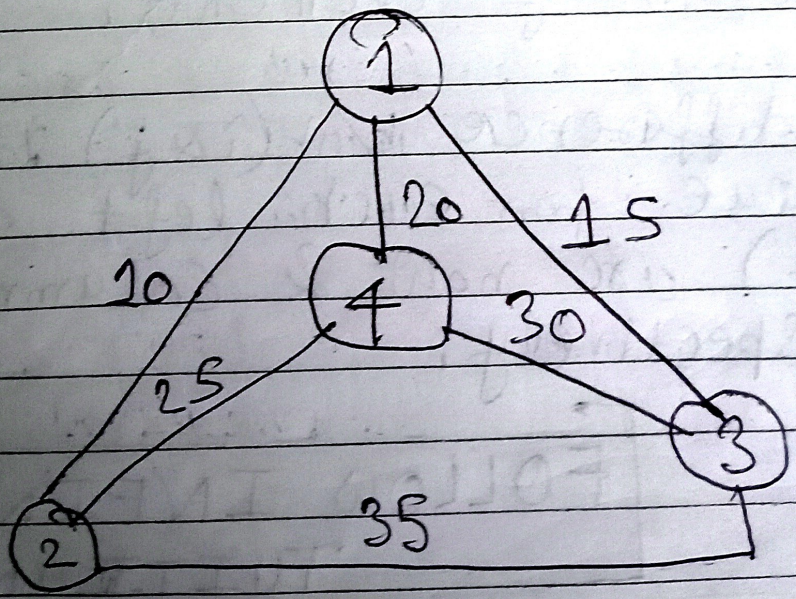
⇒ 1st Priority : → TSP → Hamilton Cycle

⇒ 2nd Priority : → Stat Space Problem.

★(1): TSP (Travelling Salesman Problem):

→ Let say a set of cities and the distance between every pair of cities, the problem is to find the shortest possible route that visits every cities exactly once & returns to the starting point.

→



Date : / /
Page No.

-> Let Consider, this graph previous. A (TSP) tour in the graph is 1-2-4-3-1. The cost of the tour is 10+25+30+15 which is 80. The problem is a famous NP- hard - problem. There is no polynomial -time Know solution of this problem.

-> Native Solution:

1) Consider city (1) as the starting & ending point.
2) Generate all (n-1)! (Permutations) of cities.
3) Calculate the ~~cost~~ cost of every permutation & keep track of the minimum cost permutation.
4) Return the permutation with minimum cost.

-> Time Complexity: $O(n^2 * 2^n)$, where $O(n^2 * 2^n)$ are maximum no. of unique subproblems/states and $O(n)$ for transition (through for loop as in code) in every state.

-> Auxillary Space: $O(n * 2^n)$, where (n) is a number of nodes/cities.

Date : / /
Page No.

☆(2) : 2nd Priority Topic :

☆ State Space Problem :

→ It is a mathematical representation of a problem that defined all possible stated that a problem can be.

→ It'd Common techniqued used in (AI) Artificial Intelligence.

→ A state space problem consist of :

1) A set of stated (S).
2) An initial state $s \in S$
3) A set of goal stated $T \subseteq S$
4) A finite set of actions $A = \{a_1, \ldots, a_m\}$

→ Each action (a) is $S \to S$, transform a state into another state.

→ The goal of state space problem is to find the path that leads from the initial to final path/state.