Tutorial 6

Dynamic Programming

Date of the Session: ....

6.1 PRE-TUTORIAL

1. What is Dynamic Programming?

Dynamic Programming is a technique that breaks the problems into sub-problems & save the result for future purposes. So that we do not need to compute the result again.

• The main use of dynamic programming is to solve optimization.

Limitations: The method is applicable to only those problems which pos.

2. State the different types of Dynamic Programming?

The different types of dynamic programming are knapsack & travelling Salesman.

6.2 IN-TUTORIAL

1 Maximize the profit for the 0/1 knapsack problem using dynamic programming when W = 10

Profit	Weight
10	5
40	4
30	6
50	3

Given $n=4$; $w=10$

$S^{i+1} = S^i \cup S^i_1$

$S^1 = S^0 \cup S^0_1$

$S^0 = \{(0,0)\}$

$S^0_1 = \{(10,5)\}$

$S^1 = \{(0,0), (10,5)\}$

$S^2 = S^1 \cup S^1_1$ • $S^1 = \{(0,0), (10,5)\}$

$S^1_1 = \{(40,4), (50,9)\}$

$S^3 = S^2 \cup S^2_1$

$S^3 = \{(30,6), (40,11), (40,4), (50,9), (30,6), (70,10)\}$

$S^4 = S^3 \cup S^3_1$

$S^4 = \{(0,0), (10,5), (40,4), (50,9), (30,6), (70,10), (50,3), (60,8), (90,7)\}$

$x^4 = 1$ $S^4 \in (90,7)$ ss $S^3 \notin (90,7)$

$x^3 = 0$ $S^3 \in (40,4)$ ss $S^2 \in (40,4)$

$x^2 = 1$ $S^2 \in (40,4)$ ss $S^1 \notin (40,4)$

$x^1 = 0$ $S^1 \in (0,0)$ ss $S^0 \in (0,0)$

$40+50 = 90$

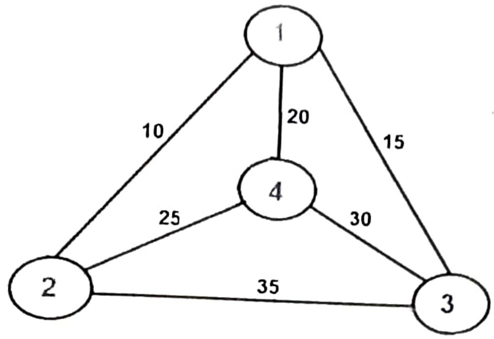
6.3 POST-TUTORIAL

1. Explain the concept of travelling salesman problem with real-time example

Solution

The travelling salesman problem (TSO) is a classic optimization problem in computer science & operations research. The goal is to find the shortest possible route that allows a salesman to visit a set of cities, exactly & return to the standing city.

2. Given a set of cities and the distance between every pair of cities, the problem is to find the shortest possible route that visits every city exactly once and returns to the starting point.



Solution:

$$C_0(V,E) = \begin{cases} 0 & \text{if } i=j \\ c_{i,j} & \text{if } (i,j) \end{cases}$$

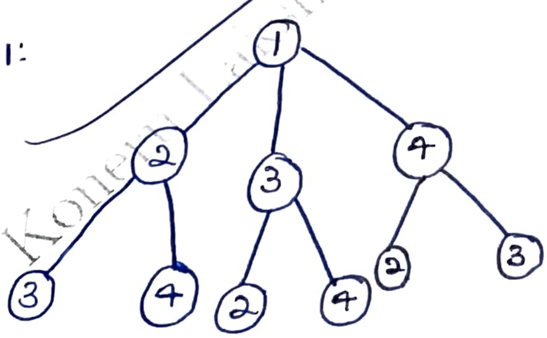
$$C(i,V) = \min\{d[i,j] + C(j, V-\{j\})\}$$

$$i \in V \ \& \ j \in V$$

Sol·1) First of all we will split the Graph ① into sub-problem.

We are choosing node-1 initial node.

step 1:



Step 2:

$$C_0\{2,\{3\},1\} = \min[d[2,3] + C(3,\phi,1)]$$

$$\min\{9+6\} = 15.$$

$$C_0\{2,\{4\},1\} = \min\{d[2,4] + C(4,\phi,1)\}$$

$$= \min\{10+8\} = 18$$

$$Co\{3, \{2\}, 1\} = \min\{d[3,2] + c\{2, \phi, 1\}\}$$

$$= \{13 + 5\} = 18$$

$$Co\{3, \{4\}, 1\} = \min\{d[3,4] + c[4, \phi, 1]\}$$

$$= 9 + 8 = 17$$

$$Co\{4, \{2\}, 1\} = \min\{d[4,2] + c[2, \phi, 1]\}$$

$$= 8 + 5 = 13$$

$$Co\{4, \{3\}, 1\} = \min\{d[4,3] + c\{3, \phi, 1\}\}$$

$$= 12 + 6$$

$$= 18$$

step 3:

$$c\{2, \{3,4\}, 1\} = \min\{d[2,3] + c\{2, \{3\}, 1\}, d[2,4] + c\{2, \{4\}, 1\}\}$$

$$= \min\{9 + 15, 10 + 18\} = 24$$

$$c\{3, \{2,4\}, 1\} = \min\{d[3,2] + c\{3, \{2\}, 1\}, d[3,4] + c\{3, \{4\}, 1\}\}$$

$$= \min\{8 + 18, 9 + 17\} = 26.$$

$$c\{4, \{2,3\}, 1\} = \min\{d[4,2] + c(4, \{2\}, 1), d[4,3] + c(4, \{3\}, 1)\}$$

$$= \min\{8 + 13, 9 + 18\} = 21$$

$$c\{1, \{2,3,4\}, 1\} = \min\{d[1,2] + c\{2, \{3,4\}, 1\}, d[1,3] + c[3, \{2,4\}, 1], d[1,4] + c[4, \{2,3\}, 1]\}$$

$$= \min\{10 + 24, 15 + 26, 20 + 21\}$$

$$= \min\{34, 41, 41\} = 36.$$

∴ min = 36

∴ The path is 1 – 2 – 3 – 4 – 1

For Evaluator's Use only

Evaluators Comments	Evaluator's Observation
	Marks Secured ...49... out of 50
	Full Name of the Evaluator:
	Signature of the Evaluator:
	Date of Evaluation: