Design And Analysts of Algorithms
Assignment No. 1

(1905334)

Q1. Explain how greedy algorithm design technique can be used to solve Travelling Salesperson problem.

Ans1 Let $G = (V, E)$ be directed graph with edge costs C_{ij}.

=> **Tour** :- A tour of G is a directed simple cycle that includes every vertex in V.

=> **Tour Cost** :- The cost of a tour is the sum of cost of the edges on the tour.

=> The travelling salesperson problem is to find a tour of minimum cost.

A Greedy Algorithm to Solve Travelling Salesperson Problem :-

Step1: Input the distance matrix,
$[D_{ij}]$ for $i \& j = 1, 2, 3, \ldots n$
where n = no. of nodes in distance network.

Step2: Randomly select a base city, let it be X & delete the column X of the distance matrix.

Step 3. Include X as first city in the tour.

Step 4. In the row X, find least undeleted matrix cell entry & identify the corresponding column, let this column be Y.

Step 5. Include Y as the next city to be visited in the tour

Step 6. Delete the column Y of distance matrix.

Step 7. Check whether all columns of the distance matrix are deleted. If yes, go to step 9, otherwise go to step 8.

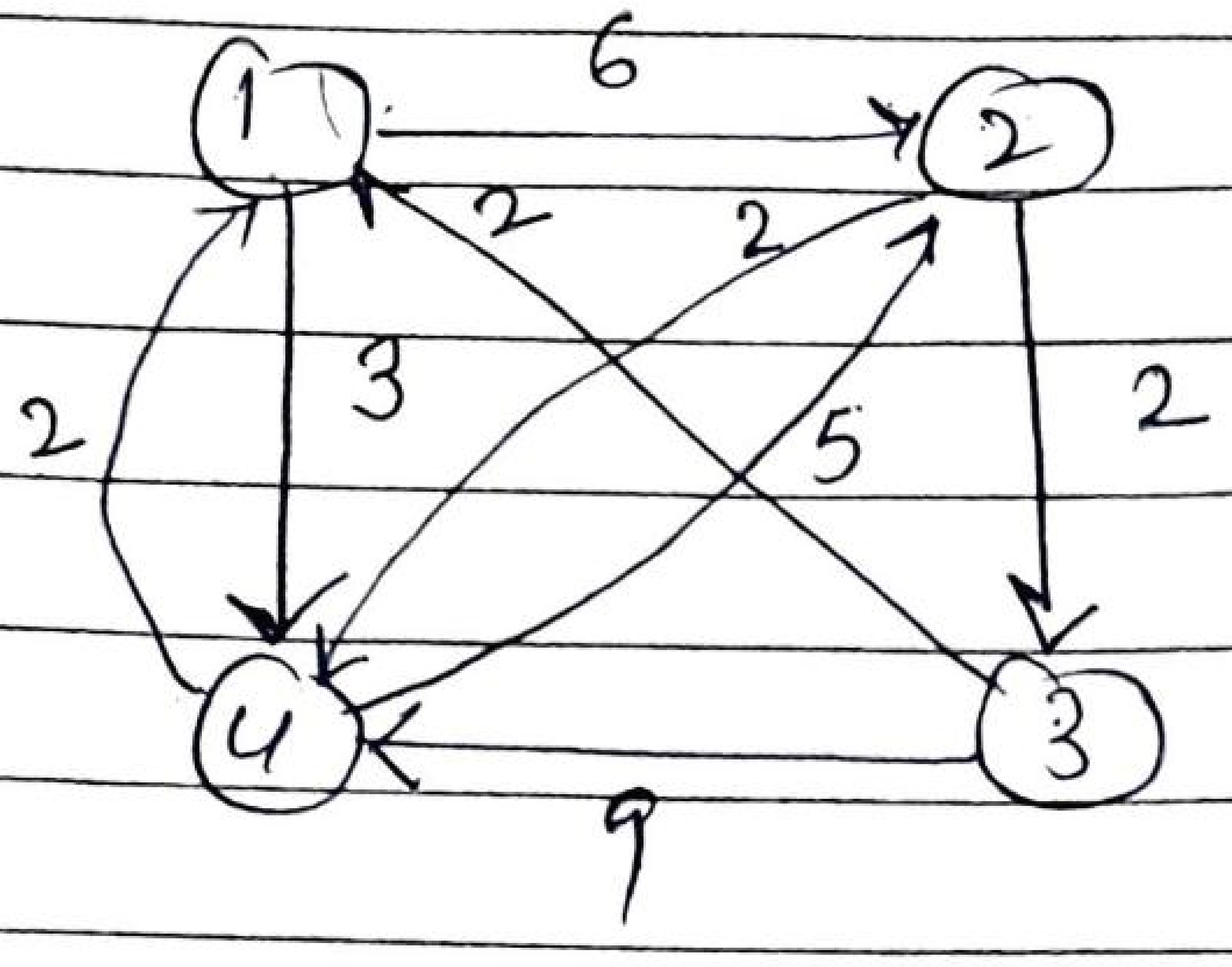
Step 8. Set X = Y, and go to step 4.

Step 9. Include the first city as the last city in the tour.

Step 10. List cities in the tour along with corresponding total distance of travel.

Time Complexity = O(n).

Example:- Apply Greedy method to solve this travelling salesperson Problem:-



Distance Matrix:-

	1	2	3	4
1	0	6	∞	3 ← least value
2	∞	0	2 ←	2 ← least value
3	2	∞	6	9
4	2	5	∞	0

least value

Tour solution:-

$$1 \xrightarrow{3} 4 \xrightarrow{5} 2 \xrightarrow{2} 3 \xrightarrow{2} 1$$

Cost = 12

First take 1 as X & 4 as Y
for next step X will be 4 & Y will be 2.

Q2. How dynamic programming can be used to solve Knapsack Problem.

Ans. The 0/1 Knapsack problem can be stated as:-

$$\text{maximize } \sum_{1 \le i \le n} p_i x_i$$ [Optimal soln & objective fnc.]

$$\text{Subject to } \sum_{1 \le i \le n} w_i x_i \le m$$ [constraint]

x_i is either 0 or 1 [feasible soln.]

=> Using dynamic programming (such that the principle of optimality holds):-

$$f_n(m) = \max\{f_{n-1}(m), f_{n-1}(m - w_n) + p_n\}$$

=> Generalizing the eqn:-

$$f_i(y) = \max\{f_{i-1}(y), f_{i-1}(y - w_i) + p_i\}$$

To represent $f_i(y)$ an ordered set s^i is used such that:-

$$s^i = \{f(y_i), y_i\}$$

Each member of s^i is a pair of (P, W)

Compute s^i, then s^i_1, s^{i+1} - - - for all elements.

Scanned by TapScanner

For example:-

$M = 8$ $\quad P = \{1, 2, 5, 6\}$

$n = 4$ $\quad W = \{2, 3, 4, 5\}$

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

$S_1^0 = \{(1,2)\}$

$S^1 = \{(0,0), (1,2)\}$

$S_1^1 = \{(2,3), (3,5)\}$

$S^2 = \{(0,0)\ (1,2),\ (2,3),\ (3,5)\}$

$S_1^2 = \{(5,0), (6,6), (7,7), \cancel{(8,9)}\}$

$S^3 = \{(0,0)\ (1,2)\ (2,3)\ \cancel{(3,5)}\ (5,4)\ (6,6)\ (7,7)\}$

$S_1^3 = \{(6,5)\ (7,7)\ (8,8)\ \cancel{(11,9)}\ \cancel{(12,11)},$
$\cancel{(13,12)}\}$

$S^4 = \{(0,0)(1,2)(2,3)(5,4)\ \cancel{(6,6)}, (6,5)$
$, (7,7), (8,8)\}$

(1) $(8,8) \in S^4$
but $(8,8) \notin S^3$ $\therefore \boxed{x_4 = 1}$
$(8-6,\ 8-5) = (2,3)$

(2) $(2,3) \in S^3$
but $(2,3) \in S^2$ $\therefore \boxed{x_3 = 0}$

(3) $(2,3) \in S^2$
but $(2,3) \notin S^1$ $\therefore \boxed{x_2 = 1}$
$(2-2,\ 3-3) = (0,0)$

(4) $(0,0) \in S^1$ and $(0,0) \in S^0$ $\therefore \boxed{x_1 = 0}$

Soln:- $\{1, 0, 1, 0\}$