TUTORIAL - VI

Ques 1

Ans 1 Minimum spanning Tree : A MST or a minimum weight spanning tree is a subset of the edges of a connected edge weighted undirected graph, that connects all the vertices together without any cycles and with the minimum possible total edge weight.

Applications :-

(i) Consider that n stations are to be linked using a communication network and laying of communication link between any two stations involves a cost. The ideal solution will be to extract a subgraph termed as minimum ~~est~~ cost spanning Tree.

(ii) Suppose you meant to construct highways or railways spanning several cities, then we can use the concept of minimum spanning tree.

(iii) Design LAN

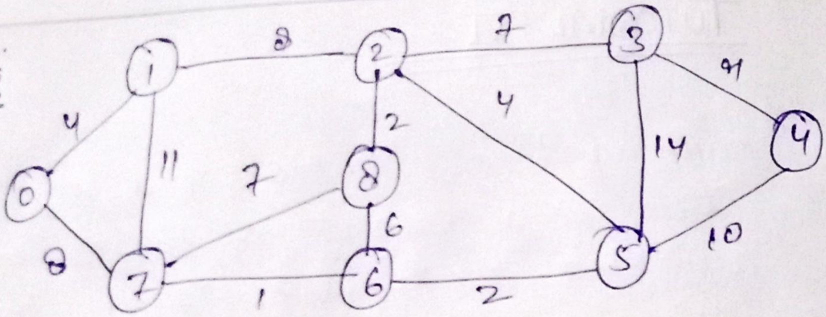
(iv) Laying pipelines connecting offshore drilling sites refineries and consumer markets.

Ques 2.

Ans 2

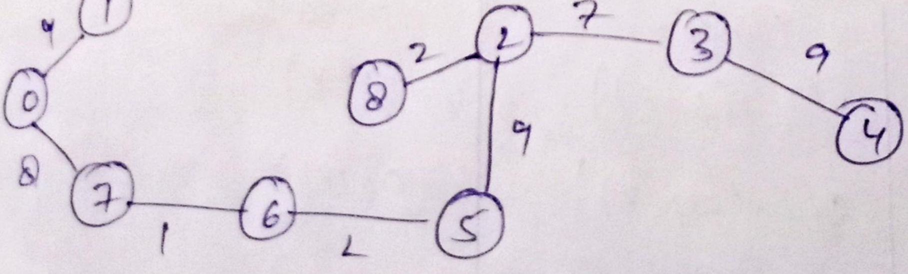
	Time	space
Prim's	$O(\log(V+E))$	$O(V)$
Kruskal's	$O(E.\log(V))$	$O(\|V\|)$
Dijkstra's	$O(V^2)$	$O(V^2)$
Bellman ford	$O(VE)$	$O(E)$

Ques 3.
Ans-3



Kruskal's Algo

U	V	W
6	7	1 ✓
5	6	2 ✓
2	8	2 ✓
0	1	4 ✓
2	5	4 ✓
6	8	6 ✗
2	3	7 ✓
7	8	7 ✗
0	7	8 ✓
1	2	8 ✗
4	3	9 ✓
4	5	10 ✗
1	7	11 ✗
3	5	14 ✗

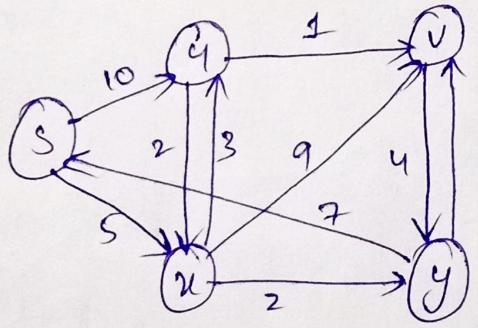


Weight $= 1+2+2+4+4+7+8+9 = 37$

Ques 4:
Ans-4 (i) The shortest path may change. The reason is there may be diff. no. of edges in diff. paths from 's' to 't' for Eg: Let shortest path be of weight 15 and has edge 5. Let there be another path with 2 edge and totot weight 25. The weight of the shortest path is increased and becomes 15+50. The weight of the other path becomes 25+20. Hence we clearly see that the shortest path changes.

(ii) If we multiply all the edges weight by 10 the shortest path doesn't change. the simple weight of all path from 's' to 't'. get multiplied by the same amount.

Ques 5.
Ans-5



Bellman Ford Algo

	s	u	v	x	y
1st →	0	10	∞	5	∞
2nd →	0	10	11	5	∞
3rd →	0	8	9	5	7
4th →	0	8	9	5	7

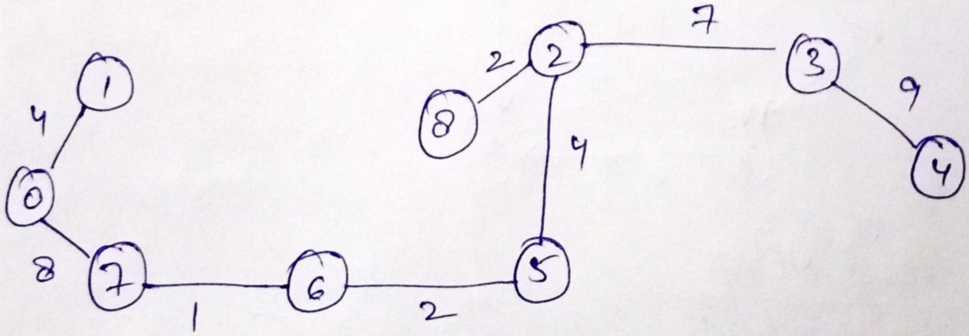
Prim's Algo

Weight :-

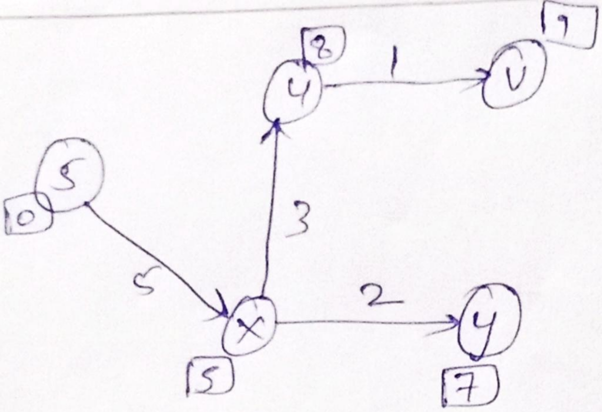
0	1	2	3	4	5	6	7	8
∅ ∞	∞	∞	∞	∞	∞	∞	∞	∞
	4						$\boxed{8}$	
		8				$\boxed{1}$		7
	11		7		4	1		$\boxed{2}$
			7		2			6
	4	14	1		10			
		$\boxed{7}$						
					$\boxed{9}$			

Parent :

0	1	2	3	4	5	6	7	8
-1	~~-1~~	~~-1~~	-1	-1	-1	~~-1~~	~~-1~~	-1
	0	1				1	1	

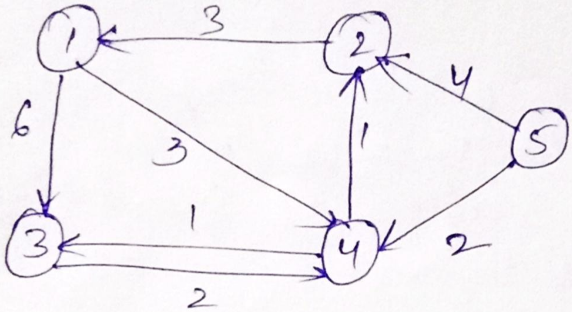


Weight :- $4+8+1+2+4+2+7+9 = 37$



<u>Ques 6.</u>

<u>Ans-6</u>



$$G_0 = \begin{array}{c c} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & \infty & 6 & 3 & \infty \\ 3 & 0 & \infty & \infty & \infty \\ \infty & \infty & 0 & 2 & \infty \\ \infty & 1 & 1 & 0 & \infty \\ \infty & 4 & \infty & 2 & 0 \end{bmatrix} \end{array}$$

$$G_1 = \begin{array}{c c} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & \infty & 6 & 3 & \infty \\ 3 & 0 & 9 & 6 & \infty \\ \infty & \infty & 0 & 2 & \infty \\ \infty & 1 & 1 & 0 & \infty \\ \infty & 4 & \infty & 2 & 0 \end{bmatrix} \end{array}$$

$$G_2 = \begin{array}{c c} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & \infty & 6 & 3 & \infty \\ 3 & 0 & 9 & 6 & \infty \\ \infty & \infty & 0 & 2 & \infty \\ 4 & 1 & 1 & 0 & \infty \\ 7 & 4 & 13 & 2 & 0 \end{bmatrix} \end{array}$$

$$G_3 = \begin{array}{c c} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & \infty & 6 & 3 & \infty \\ 3 & 0 & 9 & 6 & \infty \\ \infty & \infty & 0 & 2 & \infty \\ 4 & 1 & 1 & 0 & \infty \\ 7 & 4 & 13 & 2 & 0 \end{bmatrix} \end{array}$$

$$G_4 = \begin{array}{c c} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & 4 & 4 & 3 & \infty \\ 3 & 0 & 9 & 6 & \infty \\ 6 & 3 & 0 & 2 & \infty \\ 4 & 1 & 1 & 0 & \infty \\ 6 & 3 & 3 & 2 & 0 \end{bmatrix} \end{array}$$

$$G_5 = \begin{array}{c c} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & 4 & 4 & 3 & \infty \\ 3 & 0 & 7 & 6 & \infty \\ 6 & 3 & 0 & 2 & \infty \\ 4 & 1 & 1 & 0 & \infty \\ 6 & 3 & 2 & 2 & 0 \end{bmatrix} \end{array}$$

Node	Shortest path
u	8
x	5
v	9
y	7

Dijkska Algo

→ Create spt set which keep track of Vertices with distance. ∞.
→ We assign all the vertices with distance ∞.
then we assign dist. of Source node to 0.
→ While spt set is not include all the Vertices.

1) Pick a Vertices which is not spt set & has min. dist.
2) Include it in sptset.
3) Update ~~of the~~ dist. Value at all the adjacent Vertices of the above Vertex using Condition.

if (dist [v] > dist [u] + graph [u][v])
dist [v] = dist [u] + graph [u][v]

Node	Shortest dist
s	0
u	8
x	5
v	9
y	7