Tutorial - 6

Ans1) Minimum Spanning Tree is a subset of edges of a connected edge - weighted undirected graph that connects all the vertices together without any cycles with minimum possible edge weighted.

APPLICATIONS :-

i) Consider n stations are to be linked using a communication network and laying of communication link b/w any two stations involves a cost. The ideal solution would be to extract a subgraph termed as minimum spanning tree.

ii) Designing LAN.

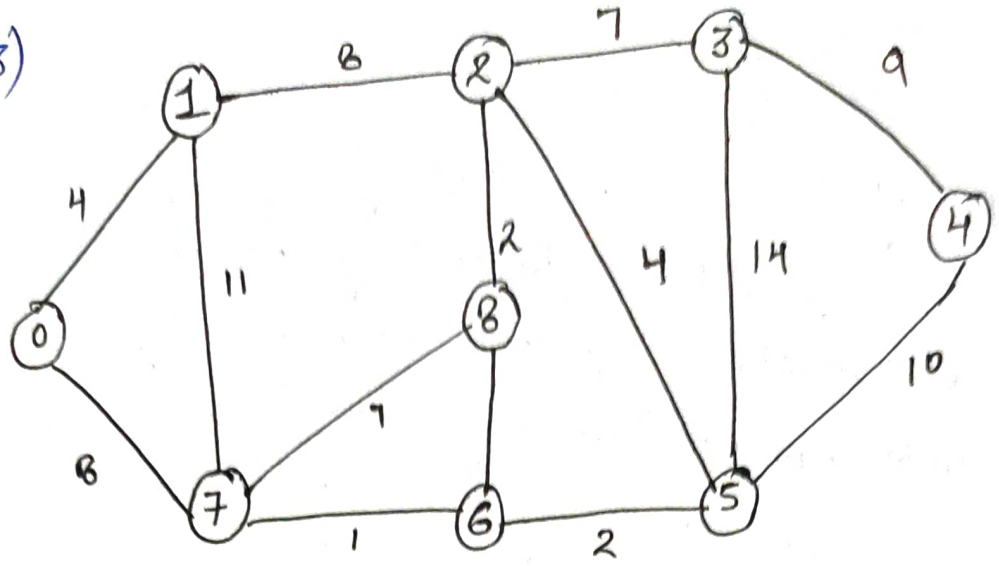
iii) Suppose you want to construct highways spanning several cities, then we can use concept of MST.

iv) Laying pipeline connecting offshore drilling sites, refineries consumer markets.

Ans2]

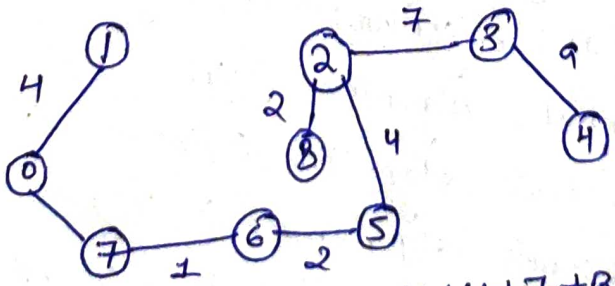
Time Complexity of Prim's Algorithm : $O(|E| \log |V|)$
Space Complexity of Prim's Algorithm : $O|V|$
Time Complexity of Kruskal's Algorithm: $O|E| \log |E|$
Space Complexity of Krushal's Algorithm : $O|V|$
Time Complexity of Dijkstra's Algorithm: $O(V^2)$
Space complexity of Dijkstra's Algorithm : $O(V^2)$
Time complexity of Bellman Ford's Algorithm : $O(VE)$
space complexity of Bellman Ford's Algorithm: $O(E)$

Ans3)



Kruskal's Algorithm

u	v	w	
0	7	1	✓
6	5	2	✓
2	8	2	✓
0	1	4	✓
2	5	4	✓
6	8	6	X
2	3	7	✓
7	8	7	X
0	7	8	✓
1	2	8	X
4	3	9	✓
4	5	10	X
1	7	11	X
3	5	14	X



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

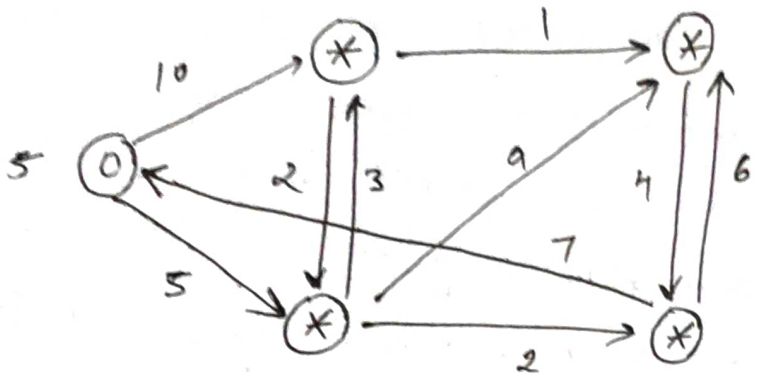
Prim's Algorithm

Weight = 4+8+2+4+2+7+9+3
= 37

Ans4) i) The shortest path may change. The reason is that there may be different no of edges in different paths from 's' to 't'. For eg- Let the shortest path of weight is and has edges s.
Let there we another path with 2 edges and total weight 2s. The weight of shortest path is increased by s*10 and becomes 1s+s0. Weight of other path is increased by 2*10 becomes 2s+20. So, the shortest path changes to other path with weight as45

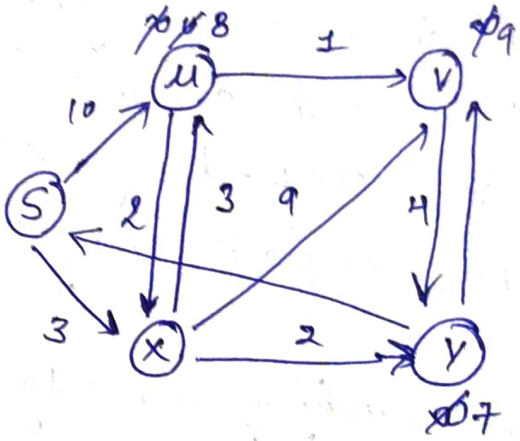
ii) If we multiply all edges weight by 10, the shortest path denote change. The reason is that weights of all path from 's' to 't' gets multiplied by same unit. The no of edges or path doesn't matter.

Ans5)



Dijkstra's Algorithm:-

NODE	SHORTEST PATH FROM SOURCE NODE
u	8
x	5
v	9
y	7



Bellman Ford Algorithm:

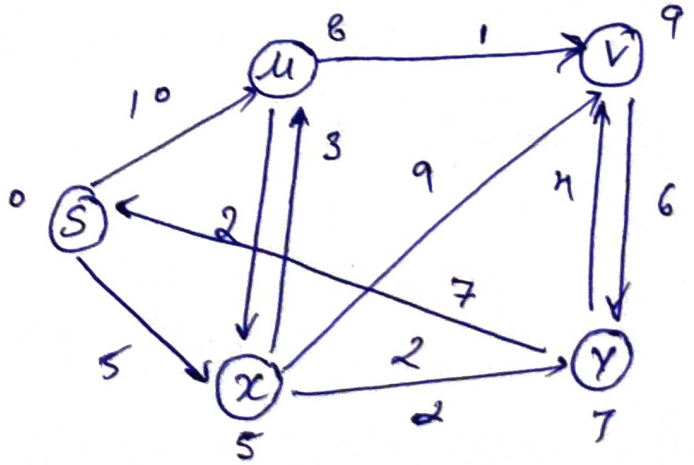
1st → (s) 0, (u) ∞, (v) ∞, (x) ∞, (y) ∞ → (s) 0, (u) ∞, (v) ∞, (x) 5, (y) ∞

2nd → (s) 0, (u) 10, (v) 11, (x) 5, (y) ∞ → (s) 0, (u) 8, (v) 9, (x) 5, (y) 7

3rd → (s) 0, (u) 8, (v) 9, (x) 5, (y) 7

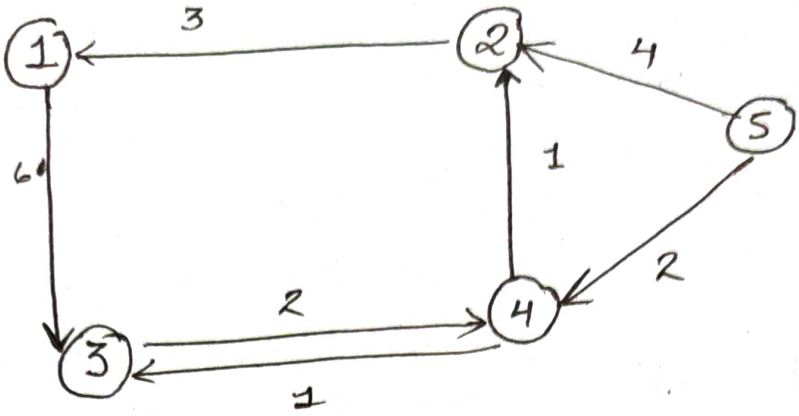
4th → (s) 0, (u) 8, (v) 9, (x) 5, (y) 7

graph does not have negative cycle.



→ Final Graph

Ans 6



	1	2	3	4	5
1	0	∞	6	3	∞
2	2	0	∞	∞	∞
3	∞	∞	0	2	∞
4	∞	1	1	0	∞
5	∞	4	∞	2	0

	1	2	3	4	5
1	0	∞	6	3	∞
2	2	0	8	5	∞
3	∞	∞	0	2	∞
4	∞	1	1	0	∞
5	∞	4	∞	2	0

	1	2	3	4	5
1	0	∞	6	3	∞
2	2	0	8	5	∞
3	∞	∞	0	2	∞
4	3	1	1	0	∞
5	6	4	12	2	0

	1	2	3	4	5
1	0	∞	6	3	∞
2	2	0	8	5	∞
3	∞	∞	0	2	∞
4	3	1	1	0	∞
5	6	4	12	2	0

Time complexity $\longrightarrow O(|V|^3)$

Space complexity $\longrightarrow O(|V|^2)$

Ans