Tutorial-6

Sol 1 :- Minimum spanning tree :- A minimum spanning tree (MST) or 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 & with the minimum possible total edge weight.

- Applications :-

(i) consider n stations are to be linked using a communication network and lying of communication link between any two stations involves a cost.
The ideal solution would be to extract a subgraph termed as minimum cost spanning tree.

(ii) suppose you want to construct highways or railroads spanning several cities then we can use the concept of minimum spanning trees.

(iii) Designing LAN.

(iv) laying pipelines connecting offshore drilling sites, refineries & consumer markets.

(v) suppose you want to apply a set of houses with
- Electric Power
- water
- Telephone lines.
- sewage lines.

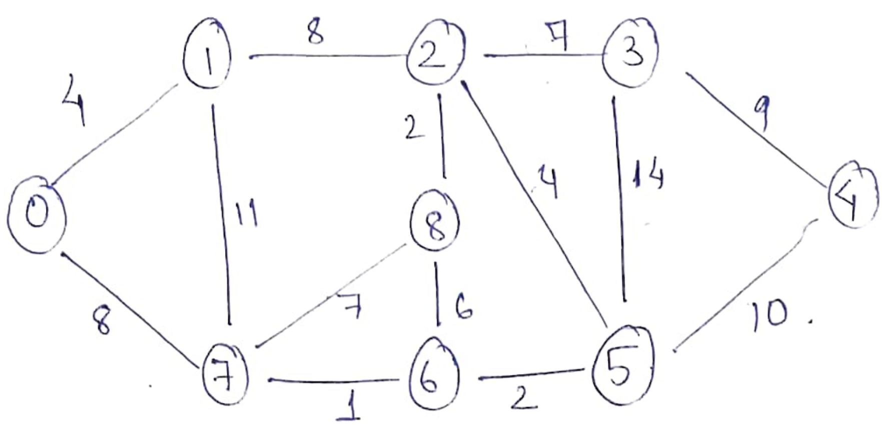
Sol2:→ Time complexity of Prim's algorithm : $O(|E| \log |V|)$
Space complexity of Prim's algorithm : $O|V|$.

→ Time complexity of Kruskal's algorithm : $O(|E| \log |E|)$
Space complexity of Kruskal'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's ford algorithm : $O(E)$

Sol3:-

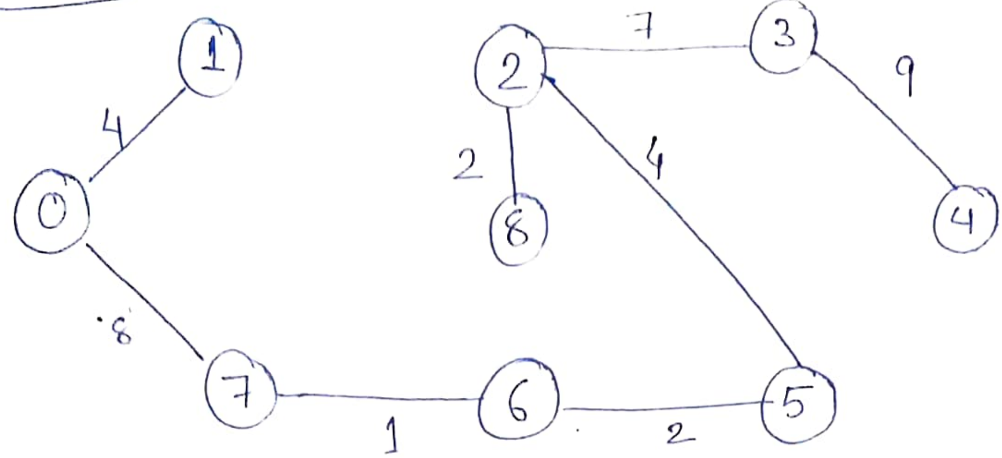


→ Kruskal's algorithm

(source) U	(dest) V	(weight) 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	✗

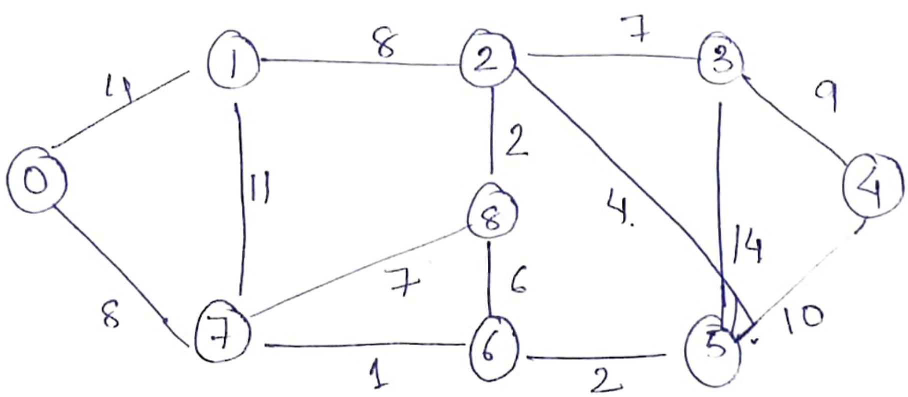
U	V	W	
3	4	9	✓
4	5	10	✗
1	7	11	✗
3	5	14	✗

Tut-6



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

Prim's Algorithm

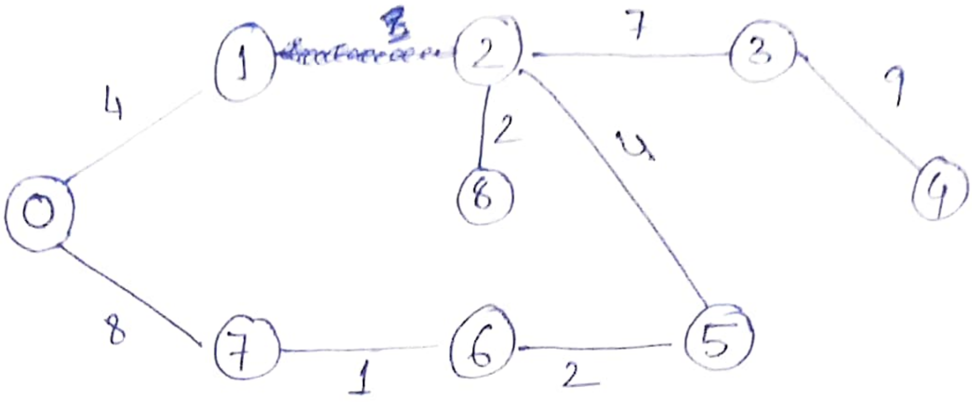


weight

0	1	2	3	4	5	6	7	8
$\boxed{0\ \cancel{\infty}}$	∞	∞	∞	∞	∞	∞	∞	∞
	$\boxed{4}$						$\boxed{8}$	
		$\boxed{8}$						
	11					$\boxed{1}$		7
			7		4			$\boxed{2}$
					$\boxed{2}$	[illegible]	[illegible]	6
		$\boxed{4}$	14	10				
			$\boxed{7}$					
				$\boxed{9}$				

Parent :-

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

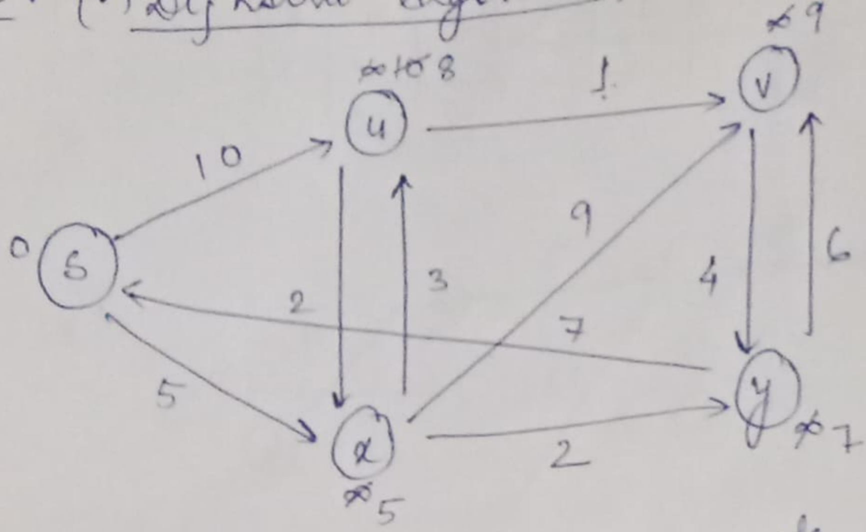


weight $= 4+8+1+2+4+2+7+9 = 37$ ans.

Sol 4:-(i) The shortest path may change. The reason is, there may be different number of edges in different paths from 's' to 't'. For example, let shortest path be of weight 15 and has edge 5 edges. Let there be another path with 2 edges & total weight 25. The weight of the shortest path is increased by 5×10 & becomes $15 + 50$. Weight of the other path is increased by 2×10 & becomes $25 + 20$. So the shortest path changes to the other path with weight as 45.

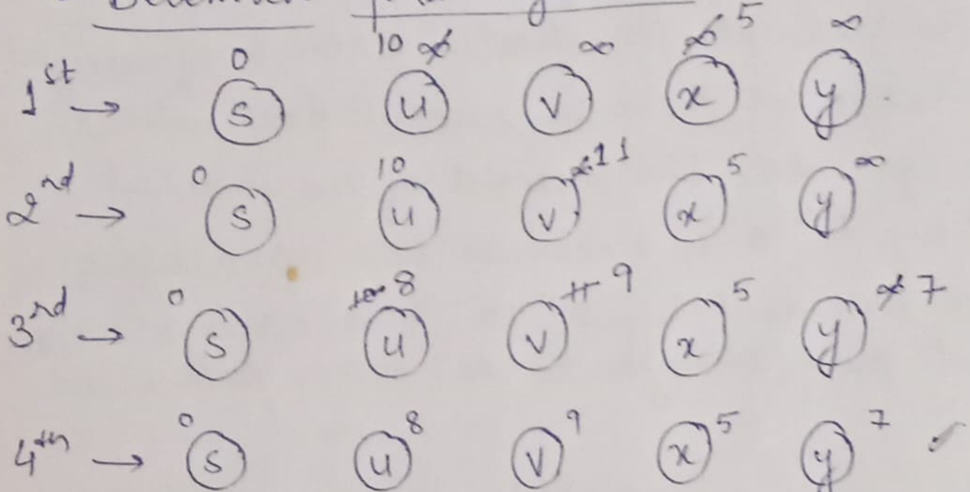
(ii) If we multiply all edges weight by 10, the shortest path doesn't change. The reason is simple, weights of all paths from 's' to 't' get multiplied by same amount. The number of edges on a path doesn't matter. It is like changing units of weights

Sol 5:- (•) Dijkstra algorithm.



node	shortest dist from source node
u	8
x	5
v	9
y	7

→ Bellman ford algorithm



graph doesnot ha[ve] -ve cycle.

final graph:-