# Tutorial Sheet - 6

Sol:- Minimum 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 weighted.

Application :-

i) Consider n stations are to be linked using a communication network and lying of ~~com~~ communication link between any two stations involves a cost.

The ideal graph solution would be to extract a subgraph termed as minimum cost spanning tree.

ii) Suppose you want to construct highways or ~~ro~~ 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 meant to apply a set of houses with:-

→ electric power.

→ Water.

→ telephone lines.

→ Sewage lines.

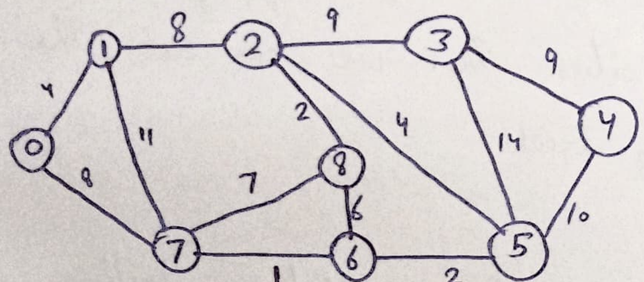
Sol 2. - Time complexity of prim's algo : $O(|E| \log|V|)$
Space complexity of prim's algo : $O(V)$

Time complexity of Kruskal's algo : $O(|E| \log |E|)$
Space complexity of Kruskal's algo : $O(|V|)$

Time complexity of Dijstra's algo : $O(V^2)$
Space complexity of Dijkstra's algo : $O(V^2)$

Time complexity of Bellman ford's algo : $O(VE)$
Space complexity of Bellman ford's algo : $O(E)$.

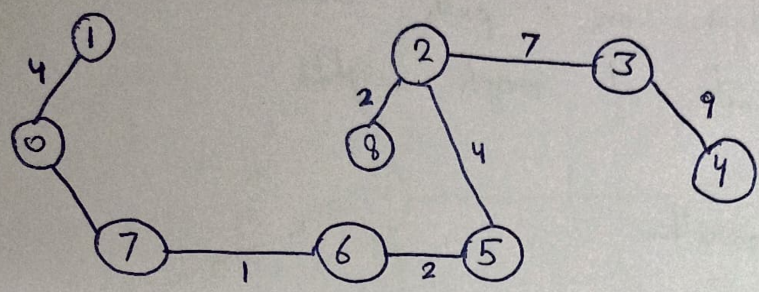
Sol 3 -




Kruskal's algo :

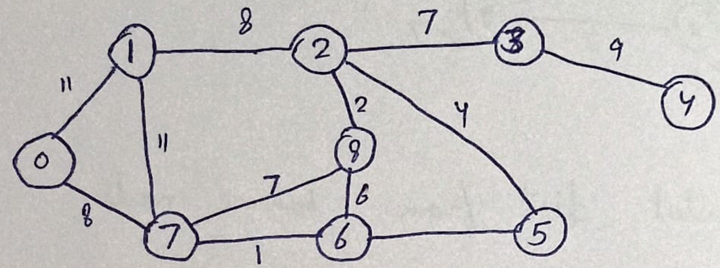
| O | V | W | |
|---|---|---|---|
| 6 | 7 | 1 | ✓ |
| 5 | 6 | 2 | ✓ |
| 2 | 8 | 2 | ✓ |
| 0 | 1 | 4 | ✓ |
| 2 | 5 | 4 | ✓ |
| 6 | 8 | 6 | × |
| 2 | 3 | 7 | ✓ |
| 7 | 8 | 7 | × |
| 0 | 7 | 8 | ✓ |
| 1 | 2 | 8 | × |

| O | V | W | |
|---|---|---|---|
| 4 | 3 | 9 | ✓ |
| 4 | 5 | 10 | × |
| 1 | 7 | 11 | × |
| 3 | 5 | 14 | × |




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

Prim's algorithm :-



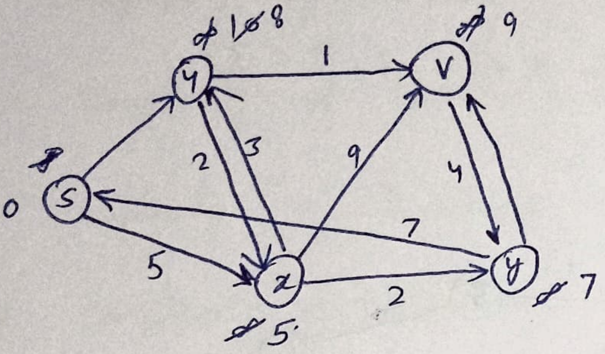

Weight = 4 + 8 + 1 + 2 + 4 + 2 + 7 + 9 = 37.

Sol 4 = (i) The shortest path may change. The reason is there may be different number of ~~ed~~ edges in different path from 's' to 't' for eg, let shortest path be of weight 15 and has edge 5 edges. Let there be another path with 2 edges and total weight 25. This weight of the shortest path is increased by $5 \times 10$ and ~~it~~ becomes 15 + 50 weight of the other path is increased by $2 \times 10$ & becomes 25 + 20. So, the shortest path changes to the other path with weight ~~d~~ as 45.

(ii) If we multiply all edges weight by 60, the ~~sorted~~ shortest path doesn't change. The reason is simple, weights of all path from 's' to 't' get multiplied by same amount.

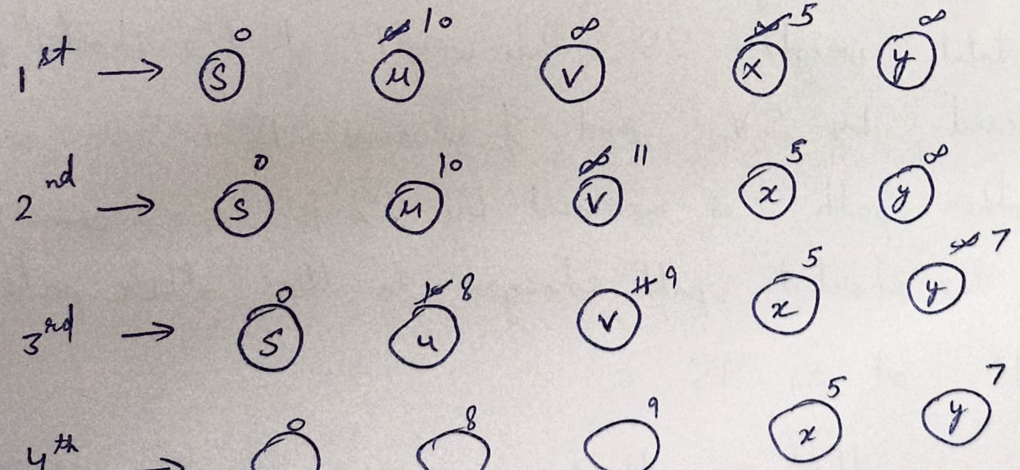
The number of edges on a path doesn't matter. It is like changing units of ~~weigh~~ weights.

Sol 5. Dijkstra Algorithm:



| Node | Shortest dist from source node. |
|---|---|
| u | 8 |
| x | 5 |
| v | 9 |
| y | 7 |

⇒ Bellman ford algo:




graph doesn't have -ve cycle.

Final graph