Name: Nishtha Jain
Section: H
Roll No.: 39



# Design And Analysis Of Algorithms (Tutorial-6)

**Q1** What do you mean by minimum spanning tree? What are the applications of MST?

⇒ Minimum Spanning Tree (MST): 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 and with the minimum possible total edge weight.

Applications:

* Consider 'n' stations are to be linked using a communication network and lying of communication link between any two station involve a cost. The ideal solution would be to exact a subgraph termed as Minimum cost Spanning Tree.
* Suppose you meant to construct highways or railroads spanning several cities then we can use the concept of minimum spanning tree.
* Design LAN.
* Laying pipelines, connecting offshore drilling sites, refineries and consumer markets.

**Q2** Please analyse the time and space complexity of Prim, Kruskal, Dijkstra and Bellman ford algorithm.

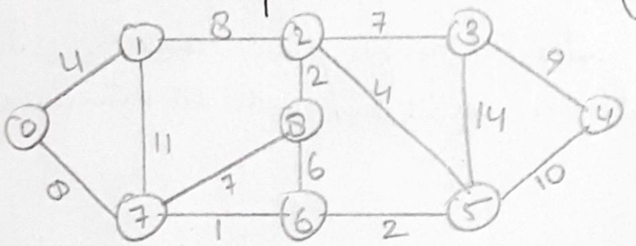
⇒ Prim's Algorithm: Time complexity ⇒ $O((V+E)\log)$
Space complexity ⇒ $O(V)$

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

Dijkstra's Algorithm : Time Complexity → $O(V^2)$
Space Complexity → $O(V^2)$

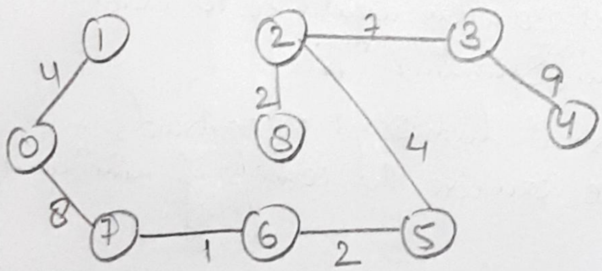
Bellman ford Space Complexity → $O(E)$
Time Complexity → $O(VE)$

**Q3** Apply Kruskal's and Prim's algorithm on graph given on right side to compute MST and its weight ?

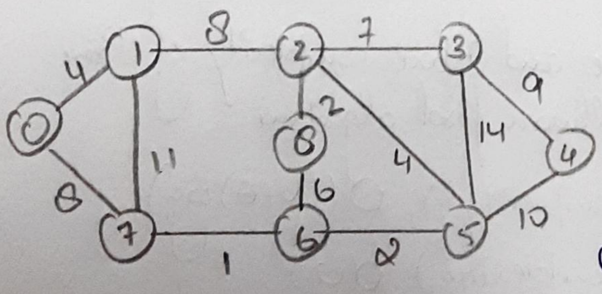


⇒ Kruskal's Algorithm :

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



Weight ⇒ 1+2+2+2+4+4+7+8+9
⇒ 39



⇒ Prim's Algorithm :

~~Weight →~~

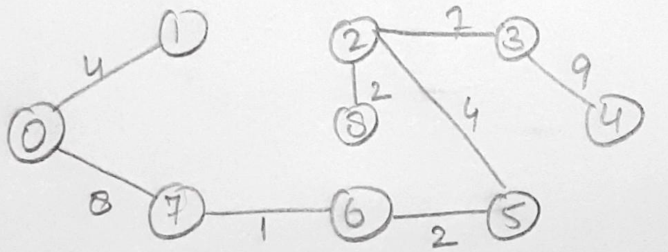
~~0 1 2 3 4 5 6 7 8~~

Weight

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

Parent :)

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




Weight :) 4+8+1+2+4+2+7+9

:) $\boxed{37}$.

Qu Given a directed weighted graph. You are also given the shortest path from source vertex 's' to a destination vertex 't'. Does the shortest path remain same in the modified graph in following cases?

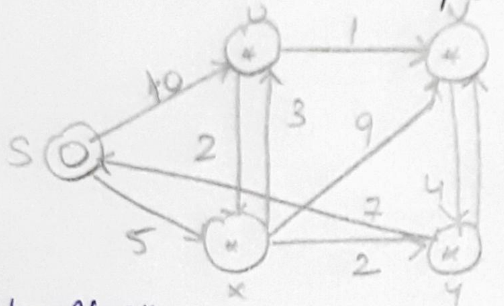
- If weight of every edge is increased by 10 units.
- If weight of every edge ~~of~~ is multiplied by 10 units.

⇒ ① The shortest path may change. The reason is, there may be different number of edges in different paths from 's' to 't'. Eg: let shortest path be of weight 15 and has edge 5. Let there be another path with 2 edge and total weight 25. The

weight of the shortest path is increased by 5 × 10 and becomes 15+50
weight of the other path is increased by 2 × 10 and becomes 25+20. So the shortest path changes to the other path with weight as 45.

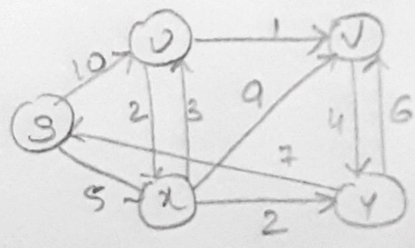
(ii) If we multiply all edges weight by 10, the shortest path don't change. The reason is simple, weight of all path from 's' to 't' be multiplied by same amount. The no. of edges on a path doesn't matter. It is like changing limits of weight.

Q5. Apply Dijkstra and Bellman algorithm on graph given on right side to compute shortest path to all nodes from node S




## Dijkstra Algorithm

| node | Shortest distance from source node |
|---|---|
| u | 8 |
| x | 5 |
| v | 9 |
| y | 7 |




## Bellman ford Algorithm

1st → S⁰ u¹⁰ v^∞ x⁵ y^∞
2nd → S⁰ u¹⁰ v^11.1 x⁵ y^∞ ⇒ Graph does not have cycle.
3rd → S⁰ u⁸ v⁹ x⁵ y⁷
4th → S⁰ u⁸ v⁹ x⁵ y⁷