Aditi Banerji
38-F

Tutorial - 6

Q1. What do you mean by Minimum Spanning tree? what are the applications of MST.

→ Minimum spanning tree is a subset of edges of a connected edge - weighted undirected graph that connects all the vertices together without any cycles and with the minimum possible total edge weighted.

* Applications :-

1) Consider n stations are to be linked using a Communication Network and lying of Communication link between any two station involves a cost.

2) The ideal solution would be to extract a subgraph termed as Minimum Cost spanning tree.

3) Suppose you want to construct highways or Railroads spanning several cities., then we can use the concept of minimum Spanning tree.

4) Designing LAN.
5) Laying pipelines connecting offshore drilling sites, refineries and consumer markets.
6) Suppose you meant to apply a set of houses with :-
Elastic Power
Water
Telephone lines
Sewage Lines

Q2. Please analyse the time and space complexity of Prims, Krushkal, Dijkstra and Bellman Ford Algorithm. (2)

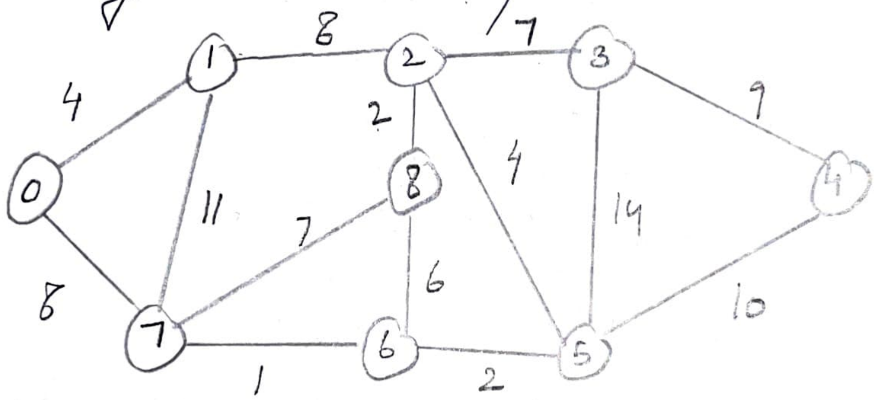
→ Time complexity of Prim's Algorithm is : $O(|E| \log|V|)$
Space Complexity of Prim's Algorithm : $O|V|$

Time complexity of Kruskal's Algorithm : $O(\log|E|)$
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 : $O(VE)$
Space Complexity of Bellman Ford : $O(E)$

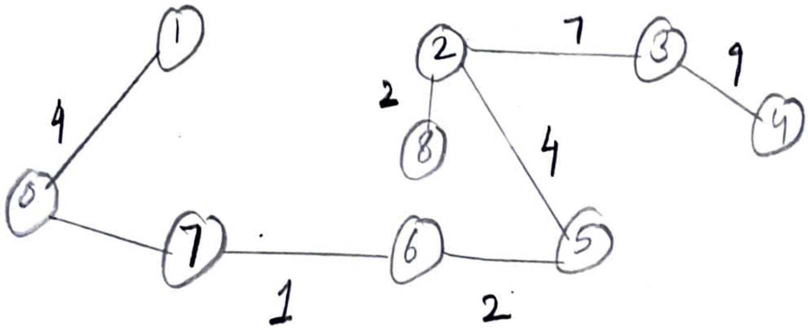
Q3. Apply Krushkal's and Prim's Algorithm on graph given on Right side to compute MST and weight?



* Krushkal's Algorithm :

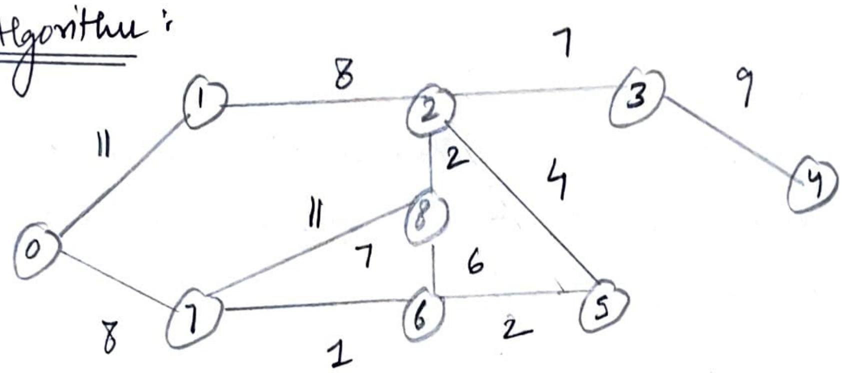
O	V	W	
6	7	1	✓
5	6	2	✓
2	8	2	✓
0	1	4	✓
2	5	4	✓
6	8	6	✗
2	3	7	✓
7	8	7	✗
0	7	8	✓
1	2	8	✗

O	W	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+2+4+2+7+9 ⇒ 37

Q4. Given a directed weighted graph. You are also given the shortest path from a source vertex 's' to a destination vertex 't'. Does the shortest path remain same in the modified graph in following case?

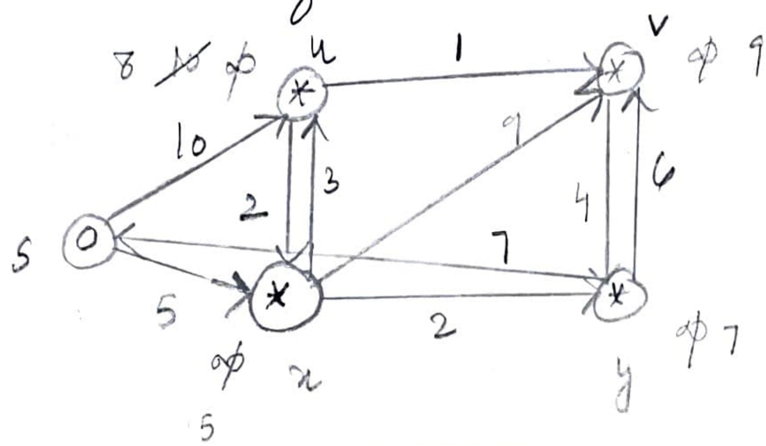
- If weight of every edge is increased by 10 units.
- If weight of every edge is multiplied by 10 units.

→ (i) The shortest path may change. The Reason is there maybe different number of edges in different paths from 's' to 't'. For eg: let shortest ~~weight~~ of path of weight 15 and has edge 5 edges.
Let there be another path with 2 edges 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 of 45.

(ii) If we multiply all edge weight by 60, the shortest path doesnot 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 doesnot matter. It is like changing units of weights.

Q5. Apply Dijkstra's and Bellman Algorithm on graph given on right side to compute shortest path to all nodes from node s.



Node	Shortest distance from Source Node
u	8
x	5
v	9
y	7

* Bellman Ford Algorithm:

1st → S^{0} $u^{\cancel{\infty}\,10}$ v^{∞} $x^{\cancel{\infty}\,5}$ y^{∞}

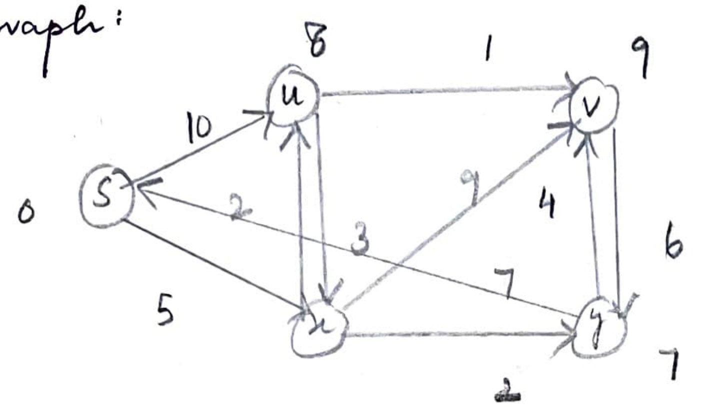
2nd → S^{0} u^{10} $v^{\cancel{\infty}\,11}$ x^{5} y^{∞}

3rd → S^{0} $u^{\cancel{\infty}\,10}$ $v^{\cancel{11}\,9}$ x^{5} $y^{\cancel{\infty}\,7}$

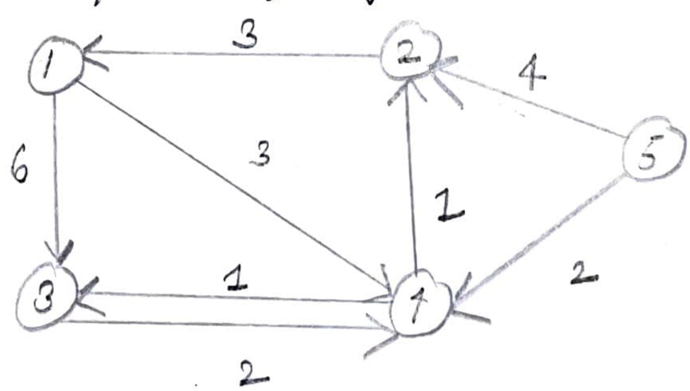
4th → S^{0} u^{8} v^{9} x^{5} y^{7}

graph doesn't have -ve cycle.

• final graph:



Q6. Apply all pair shortest path Algorithm - Floyd warshall on below mentioned graph and also analyse the time and space complexity of Algorithm.



→ Applying Floyd Warshall Algorithm

$$\begin{array}{c|c} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{bmatrix} 0 & \infty & 6 & 3 & \infty \\ 2 & 0 & \infty & \infty & \infty \\ \infty & \infty & 0 & 2 & \infty \\ \infty & 1 & 1 & 0 & \infty \\ \infty & 4 & \infty & 2 & 0 \end{bmatrix} \end{array}$$

$$\begin{array}{c|c} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{bmatrix} 0 & \infty & 6 & 3 & \infty \\ 2 & 0 & 7 & 5 & \infty \\ \infty & \infty & 0 & 2 & \infty \\ \infty & 1 & 1 & 0 & \infty \\ \infty & 4 & \infty & 2 & 2 \end{bmatrix} \end{array}$$

$$\begin{array}{c|c} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{bmatrix} 0 & \infty & 6 & 3 & \infty \\ 2 & 0 & 8 & 5 & \infty \\ \infty & \infty & 0 & 2 & \infty \\ 3 & 1 & 1 & 0 & \infty \\ 6 & 4 & 12 & 2 & 0 \end{bmatrix} \end{array}$$

$$\begin{array}{c|c} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{bmatrix} 0 & \infty & 6 & 3 & \infty \\ 2 & 0 & 8 & 5 & \infty \\ \infty & \infty & 0 & 2 & \infty \\ 3 & 1 & 1 & 0 & \infty \\ 6 & 4 & 12 & 2 & 0 \end{bmatrix} \end{array}$$

Time Complexity : $O(|V|^3)$
Space Complexity : $O(|V|^2)$