Tutorial - 6

Q1. what do you mean by minimum spanning tree? what are appliction of MST?

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

→ Applications - ① MST are used to find easy paths in maps.

② MST are used for designing networks such as water supply network or any electrical grid networks.

③ MST are used for learning salient features for real time face verification.

Q.2. Analyze the time & space complexity of Prim, Kruskal, Djikstra and Bellman Ford algorithm.

→ Bellman Ford -

- Time complexity - Best case - $O(E)$

  Avg case - $O(EV)$

  Worst case - $O(V^3)$

where V is vertices of graph and E is edges of graph

Space complexity - $O(V)$.

→ Kruskal's algorithm →

- Time complexity → Best case → $O(E(\log(E)))$

  Avg case → $O(E(\log E))$

  worst case → $O(E(\log E))$

- Space complexity - $O(E+V)$

  where E is edges and V is verties of graph.

→ Dijkstra's Algorithm -

- Time complexity → Naive :- •

  Best case - $O(V^2)$

  Avg case - $O(V^2)$

  worst case - $O(V^2)$

- Binary heap → Best case - $O(E \log V)$

  Avg case - $O(E V \log(E/V) \log V)$

  Worst case - $O(E \log V)$

- Fibonacci heap → Best case - $O(E + V \log V)$

  Avg case - $O(E+V \log(E/V) V \log V)$

  worst case - $O(E + V \log V)$

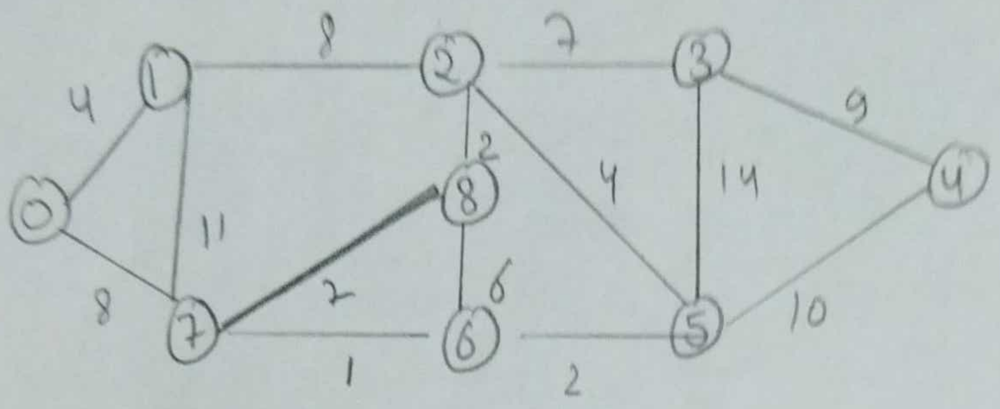
- Space complexity → $O(V)$.

→ Prim's algorithm -

- Time complexity - Best case - $O((V+E)\log V)$
  Avg case - $O((V+E)\log V)$
  worst case - $O(V^2)$

- Space complexity - $O(V+E)$.

Q.3. Apply Kruskal's and Prism algorithm for following

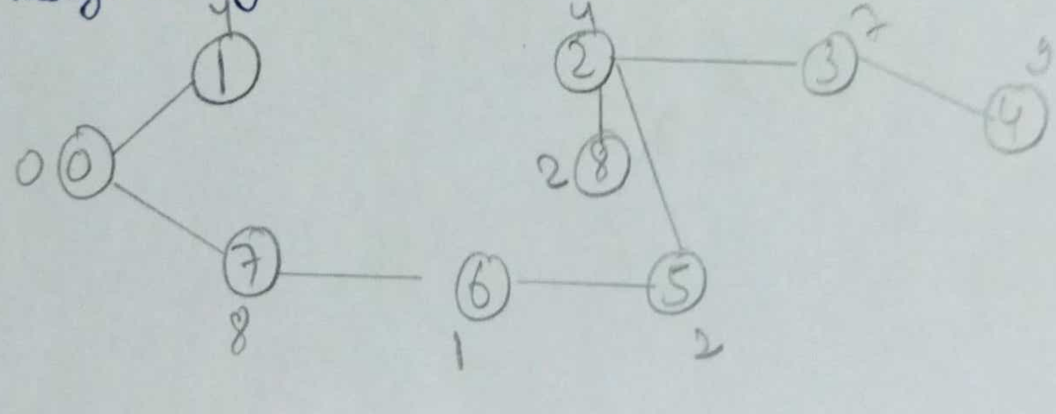


→ Prim's algorithm -

If $(w(u,v) < v.key)$
$v.key = w(u,v)$

Vertex	0	1	2	3	4	5	6	7	8
Key	0	4	4	7	9	2	1	8	2

weight of minimum spanning tree = 37

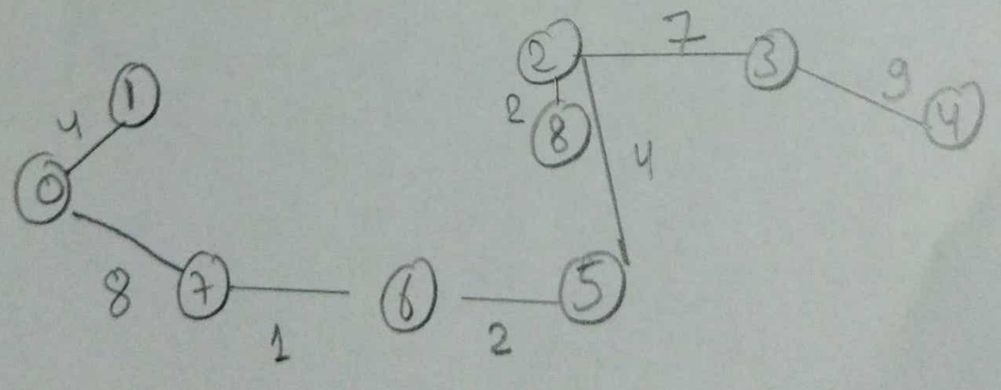


→ Kruskal's algorithm -

Finding minimum paths:

(6,7) = 1 ✓
(2,8) = 2 ✓
(6,5) = 2 ✓
(0,1) = 4 ✓
(2,5) = 4 ✓
(8,6) = 6 ×
(7,8) = 7 ×
(2,3) = 7 ✓
(0,7) = 8 ✓
(1,2) = 8 ×
(3,4) = 9 ✓
(4,5) = 10 ×
(1,7) = 11 ×
(3,5) = 14 ×

Minimum spanning tree weight = 37



Q4. 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 modified graph in following cases?

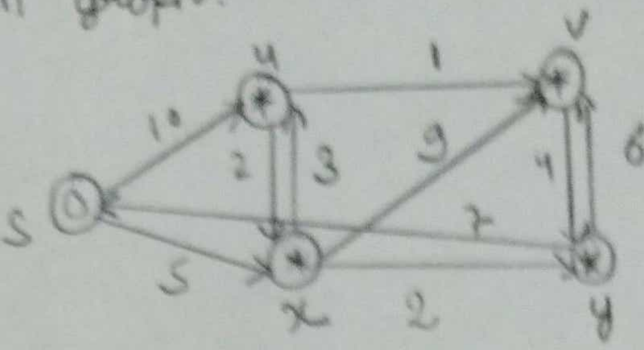
- If weight of every edge is increased by 10 units.
- If weight of every edge is multiplied by 10 units.

→ If weight of every edge is increased by 10 units the shortest path may change. The reason is there may be different number of edges in different paths from s to t.

Ex:- let shortest path is 15 of 5 edges. let there be another path with 2 edges and total weight 25. The weight of shortest path is increased by 5 * 10 and becomes 15+50. The weight of other path is increased by 2 * 10 and becomes 25+20. So the shortest path changes to other path with weight 45.

→ If we multiply all edge weights by 10 the shortest path doesn't change. The reason is weight of all paths from s to t get multiplied by same amount. The no. of edge doesn't matter. It is like changing unit of weights.

Q5: Apply Dijkstra and Bellman Ford algorithm on given graph:



Dijkstra -

s	u	v	x	y
0	∞	∞	∞	∞
0	10	5	~~∞~~ 14	~~∞~~ 7
0	~~10~~	5		
0	8	5	14	7
0	8	5	~~13~~	7
0	8	5	9	7

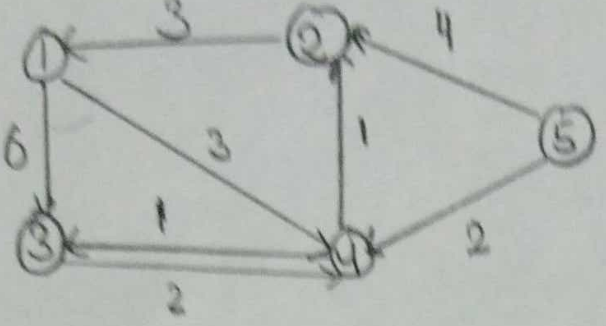
s = 0 , u = 8, v = 5, x = 9 , y = 7

Bellman Ford -

s	u	v	x	y
0	∞	∞	∞	∞
0	~~10~~	9	5	13
0	8	9	5	7

s = 0, u = 8 , v = 5, x = 9 , y = 7

Q6. Apply Floyd warshall algorithm :



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

$$D^1 = \begin{array}{c|c} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{bmatrix} 0 & 4 & 6 & 3 & \infty \\ 3 & 0 & 9 & 6 & \infty \\ 6 & 3 & 0 & 2 & \infty \\ 4 & 1 & 1 & 0 & \infty \\ 7 & 4 & 3 & 2 & 0 \end{bmatrix} \end{array}$$

$(1,2) = \infty$, $(2,3) = \infty$, $(2,4) = \infty$, $(3,1) = \infty$

$1 \to 4 \to 2$; $2 \to 1 \to 3$; $2 \to 1 \to 4$; $3 \to 4 \to 2 \to 1$

$\Rightarrow 3+1 = 4$; $\Rightarrow 3+6 = 9$; $\Rightarrow 3+3 = 6$; $\Rightarrow 2+1+3 = 6$

$(3,2) = \infty$, $(4,1) = \infty$, $(5,1) = \infty$, $(5,3) = \infty$

$3 \to 4 \to 2$; $4 \to 2 \to 1$; $5 \to 2 \to 1$; $5 \to 4 \to 3$

$\Rightarrow 2+1 = 3$; $1+3 = 4$; $4+3 = 7$; $2+1 = 3$

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

$(1,3) = 6$, $(2,3) = 9$, $(5,1) = 7$, $(5,2) = 4$

$1 \to 4 \to 3$; $2 \to 1 \to 4 \to 3$; $5 \to 4 \to 2 \to 1$; $5 \to 4 \to 2$

$= 3+2 = 5$; $3+3+3+1 = 7$; $2+1+3 = 6$; $2+1 = 3$