Design and Analysis of Algo
Assignment

1) What do you mean by minimal spanning tree? What are the applications of MST?

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

Applications of MST:

1) Network design:
MSTs are used in designing efficient communication networks, such as laying down cables for telecommunication networks or designing computer networks.

2) Cluster analysis:
MSTs are used in clustering algorithms algorithms to identify natural groups in data.

3) Approximation algorithm:
MSTs can be used as part of approximation Algo for optimization problems such as travelling Salesman Problem.

4) Image segmentation:
MSTs are used in image processing for segmentation images into regions with similar characteristics.

5) Circuit design:
In electronic circuit design, MSTs can be used to find the shortest paths connecting various components, minimizing total length of wires or traces.

Q2:- please analyse the time & space complexity of prim, Kruskal, Dijkstra & Bellman ford algorithm.

① Prim's Algo

Time complexity $O(V^2)$ → with matrix. $O((V+E)\log V)$ with adjacency list & binary heap.

Space complexity $O(V)$ for storing key & parent pointer

② Kruskal's Algo

Time complexity $O(E \log E)$ or $O(E \log V)$ with disjoint set Data structure

Space complexity $O(V+E)$ for storing graph & disjoint set DS.

③ Dijkstra's Algo

Time complexity - $O((V+E)\log V)$ with binary heap.
$O(V^2)$ → with array.

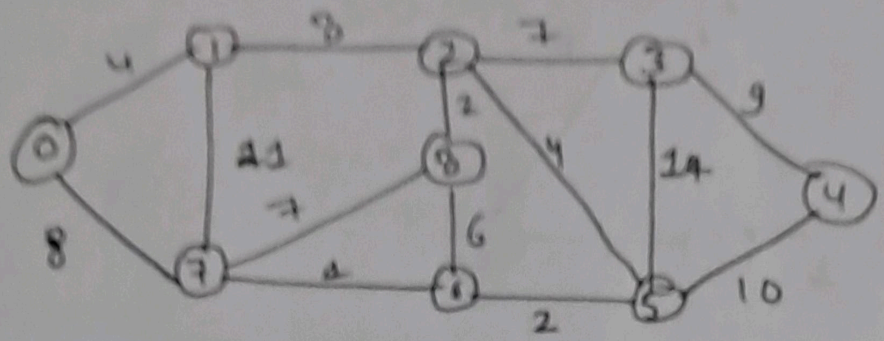
Space complexity - $O(V)$ for storing distance & parent pointer
$O(E)$ → for priority queue or heap.

④ Belman-ford Algo

Time complexity - $O(VE)$ in worst case

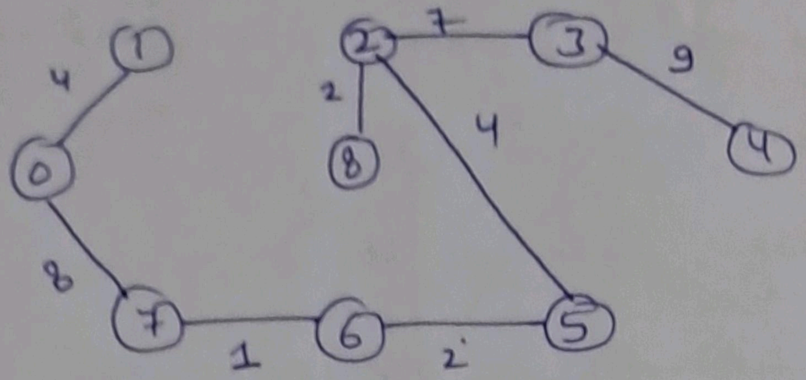
Space complexity - $O(V)$ for storing distance & parent pointer

Q3: Apply Kruskal & prim's Algo on graph given on right side to compute MST & its wt.

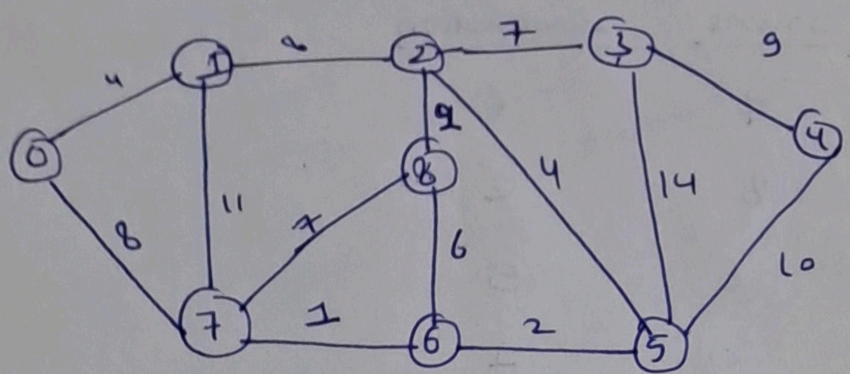


Kruskal Algo

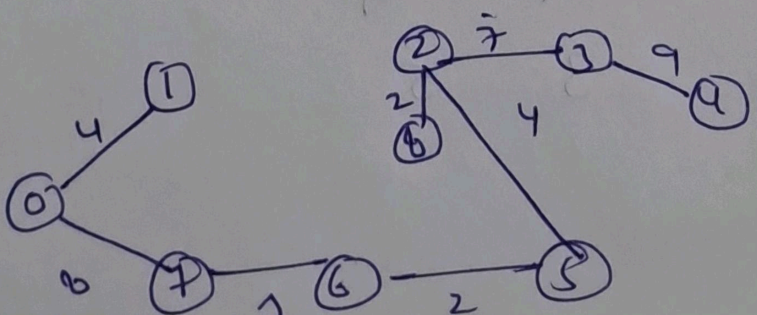
weight	Source	destination
1	7	6
2	8	2
2	6	5
4	0	1
4	2	5
6	8	6
7	2	3
7	7	8
8	0	7
8	1	2
9	3	4
10	5	4
11	1	7
14	3	5



Prism's Algo.



0	1	2	3	4	5	6	7	8
$\boxed{0}$	∞	∞	∞	∞	∞	∞	∞	∞
	$\boxed{4}$	∞	∞	∞	∞	∞	~~8~~	∞
		8	∞	∞	∞	~~∞~~	$\boxed{8}$	~~∞~~
		~~8~~	∞	∞	~~∞~~	$\boxed{1}$		~~7~~
		~~8~~	~~∞~~	∞	$\boxed{2}$			6
		$\boxed{4}$	~~14~~	∞				~~6~~
			$\boxed{7}$	~~∞~~				$\boxed{2}$
				9				



Q4: Given a directed weight 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?

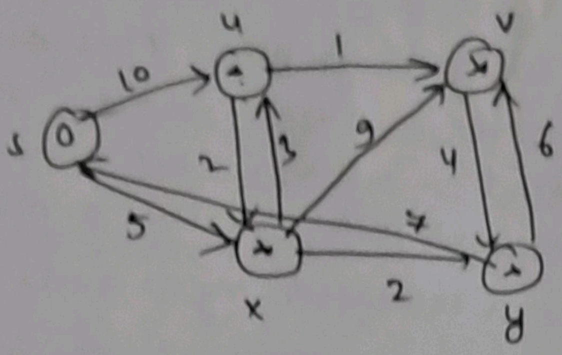
- If wt of every edge is increased by 10 units.
- If wt of every edge is multiplied by 10 units.

ans: The shortest path may change. The reason is, there may be different no. of edges in different paths from s to t.

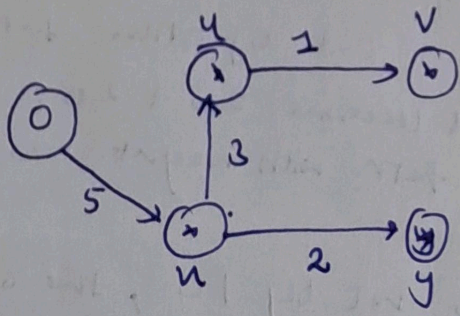
Ex: Let shortest path be of wt 15 and has 5 edges. Let another path with 2 edges & total wt 25. The wt of shortest path is increased by 5*10 & become 15+50. Wt of other path is increased by 2*10 & become 25+20. So, shortest path change to other path with weight as 45.

If we multiply all edges wt by 10, the shortest path cannot change. The reason is simple, wt of all paths from s to t get multiplied by same amount. The no. of edges on a path doesn't matter. It is like changing unit of wts.

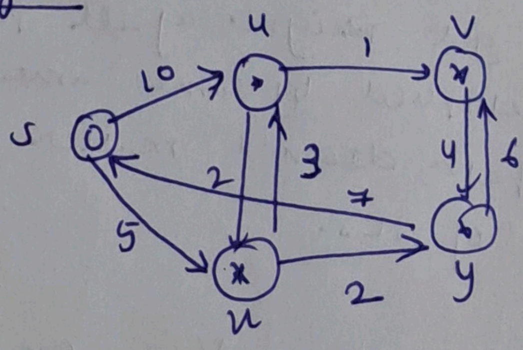
Q5: Apply Dijkstra & Belman Algo on graph given on right side to compute shortest path to all nodes from node S.



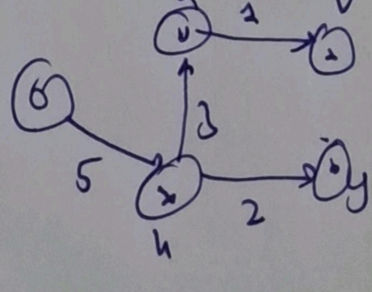
node	Shortest distance from source (0)
u	8 (5+3)
v	9 (5+3+1)
x	5
y	7 (5+2)



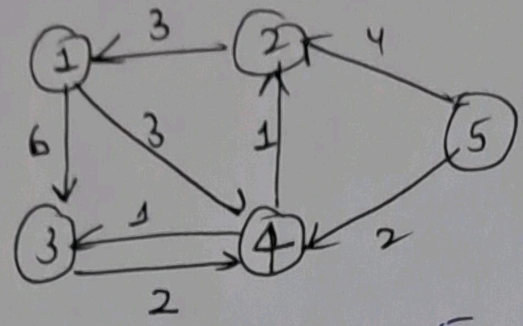
Bellman Algo



s	u	v	x	y
0	~~∞~~ ~~10~~ 3	~~∞~~ ~~6~~ 1	~~∞~~ ~~5~~ 2	~~∞~~ 2

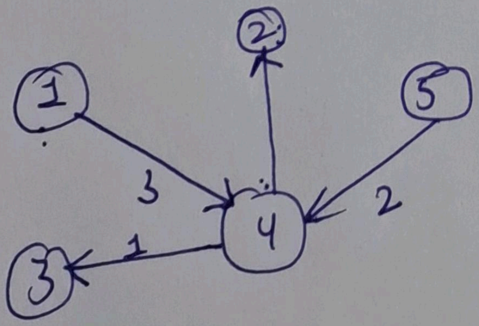


Q6:- Apply all pair shortest path Algo - Floyd Warshall on below mentioned graph after and also analyse the time + space complexity of Algo.



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

$$
\begin{array}{c|ccccc}
 & 1 & 2 & 3 & 4 & 5 \\
\hline
1 & 0 & 4 & 6 & 3 & \infty \\
2 & 3 & 0 & 9 & 6 & \infty \\
3 & 6 & 3 & 0 & 2 & \infty \\
4 & 4 & 1 & 1 & 0 & \infty \\
5 & 7 & 4 & 3 & 2 & 0
\end{array}
$$



Time complexity - $O(V^3)$

Space - $O(1)$