TUTORIAL SHEET - 6

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

Ans Minimum Spanning Tree or MST is a minimum weight spanning tree which 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 weight.

Applications

i) Consider n stations are to be linked using a communication network and laying of communication link between any two stations involves a cost. The ideal solution would be to extract a subgraph turned as minimum cost spanning tree.

ii) Suppose you want to construct highway or railroads spanning several cities then we can use the concept of minimum spanning trees.

iii) Designing LAN

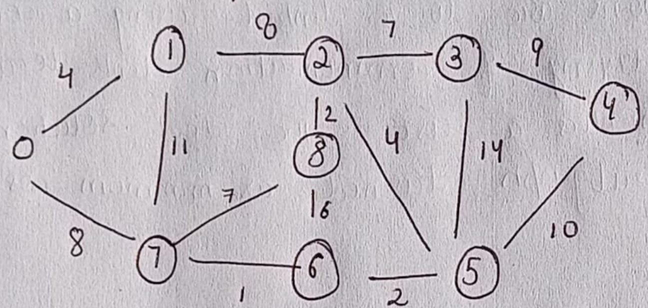
iv) Laying pipelines connecting offshore drilling sites, refineries and consumer markets.

v) Suppose you want to apply a set of houses with electric power - water - Telephone lines - Sewage line.

Que 2) Analyse the time and space complexity of Prim, Kruskal, Dijkstra and Bellman ford algorithm.

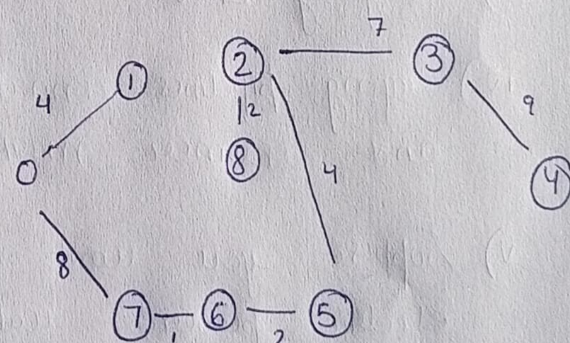
	Time Complexity	Space Complexity
Prim's Algorithm	$O(\lvert E\rvert \log \lvert V\rvert)$	$O(\lvert V\rvert)$
Kruskal's Algorithm	$O(\lvert E\rvert \log \lvert E\rvert)$	$O(\lvert V\rvert)$
Dijkstra's Algorithm	$O(V^2)$	$O(V^2)$
Bellman ford's Algorithm	$O(VE)$	$O(E)$

Que3) Apply Kruskal & Prim's Algorithm on graph given on right side to compute MST and its weight.



Kruskal's Algorithm

Source(u)	Destination(v)	Weight(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 ×
3	4	9✓
4	5	10 ×
1	7	11 ×
3	5	14 ×

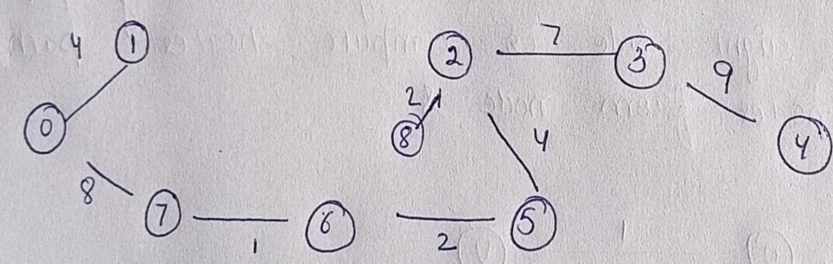


Weight = 1+2+2+4+4+7+8+9 = 37

Prim's Algorithm

Weight	0	1	2	3	4	5	6	7	8
	0 ~~∞~~	∞	∞	∞	∞	∞	∞	∞	∞
		$\boxed{4}$							
			$\boxed{8}$					$\boxed{8}$	
		11					$\boxed{1}$		7
				7		4			
						$\boxed{2}$			6
			$\boxed{4}$	14	10				
				$\boxed{7}$					
					$\boxed{9}$				

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



Que 4) 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 cases?

i) 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'. Example - Let shortest path be of weight 15 and has edge 5. Let there be another path with 2 edges and total weight 25. The

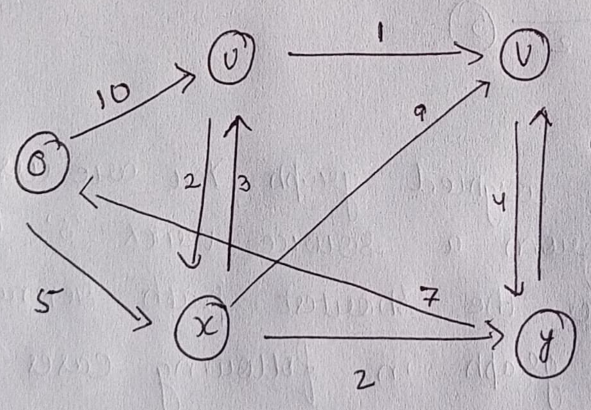
weight of the shortest path is increased by 5. 10 & becomes 15 + 50. Weight of the other pair is increased by 2 * 10 & becomes 25+20. So, the shortest path changes to the other path with weight as 45.

(ii) If weight of every edge is multiplied by 10 units

If we multiply all edges weight by 10, the shortest path doesn't change.

The reason is simple, weighs 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 weights.

Que 5) Apply Dijkstra and Bellman Algorithm on graph given on right side to compute shortest path to all nodes from node S.



Dijkstra's Algorithm

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

Bellman ford algorithm

	S	u	v	x	y
1st	0	~~∞~~ 10	∞	~~∞~~ 5	∞
2nd	0	10	~~∞~~ 11	5	∞
3rd	0	8	~~11~~ 9	5	~~∞~~ 7
4th	0	8	9	5	7

graph doesn't have -ve cycle