Tutorial Sheet--6

Date: / /

1- 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 & with the minimum possible total edge weight.

Applications

i- Consider n stations are to be linked using a communication network & laying of communication link between any two stations involves a cost. The ideal solution would be to extract a subgraph termed as min. cost spanning tree.

ii- Suppose you want to construct highway or railroads spanning several cities then we can use the concept of mini. spanning tree.

iii- Designing LAN

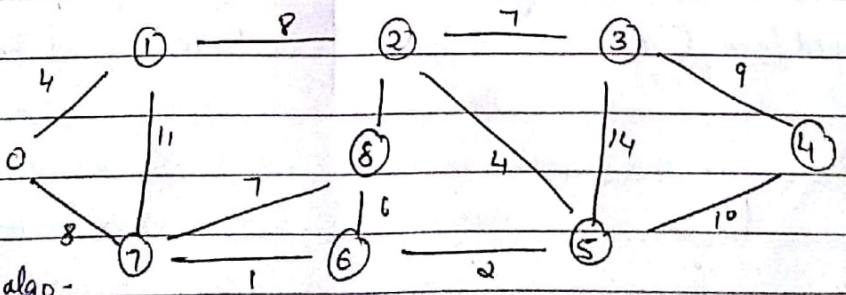
iv- Laying pipelines connecting offshore drilling sites, refineries & consumer markets.

v- Suppose you want to apply a set of houses with - Electric Power - water - Telephone lines - Sewage lines.

2-

	Time Complexity	Space Complexity
Prim's algorithm :	$O(\|E\| \log \|V\|)$	$O(\|V\|)$
Kruskal's algorithm :	$O(\|E\| \log (E))$	$O(\\|V\\|)$
Dijkstra's algorithm :	$O(V^2)$	$O(V^2)$
Bellman ford's algorithm :	$O(VE)$	$O(E)$

3-



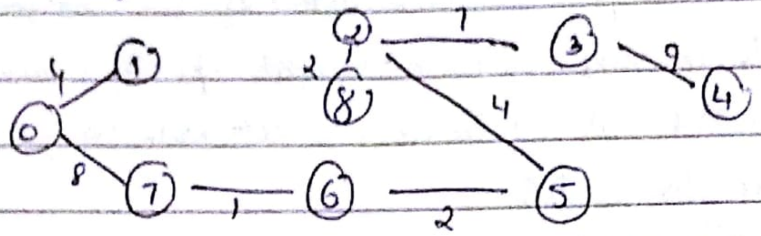
Kruskal's algo-

Source (0)	dest. (v)	weight (w)
6	7	1
5	6	2
2	8	2
0	1	4

Date: / /

u	v	w	
2	5	4	✓
6	8	6	×
2	3	7	✓
7	8	7	×
0	7	8	✓

u	v	w	
1	2	8	×
4	3	9	✓
4	5	10	×
1	7	11	×
3	5	14	×



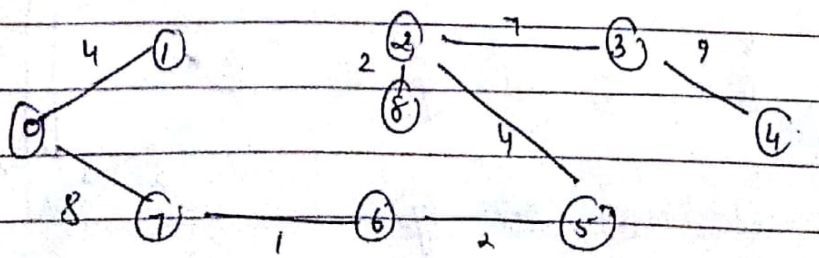
weight = 1+2+2+4+4+7+8+9 = 37

Prism's algorithm

weight –

0	1	2	3	4	5	6	7	8
[0] ∞	∞	∞	∞	∞	∞	∞	∞	∞
	[4]							
		[8]					[8]	
	11					[1]		7
			7		4			
					[2]			6
		[4]	14	10				
			[7]					
				[9]				

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



weight = 4+8+1+2+4+2+7+9 = 37.

4-

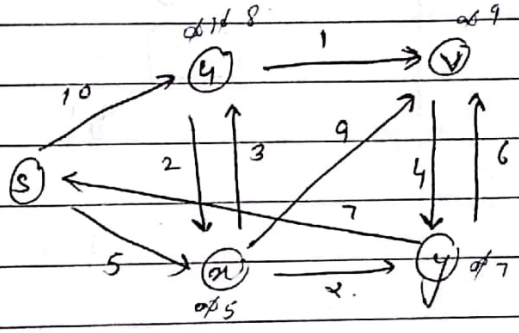
i) The shortest path may change. The reason is there may be different number of edges in different paths from 's' to 't'. Ex, let shortest path be of weight 15 & has edges 5. Let there be another path with 2 edges & total weight 25. The weight of the shortest path is increased by 5*10 & becomes 15+50. Weight of the other path is increased by 2*10 & 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 doesn't change. The reason is simple, weights of all path from 's' to 't' get multiplied by same amount. The no. of edges on a path doesn't matter. It is like changing units of weights.

Dijkstra algo:

5-

Node	Shortest dist. from source node
u	8
x	5
v	9
y	7



Bellman ford algo:

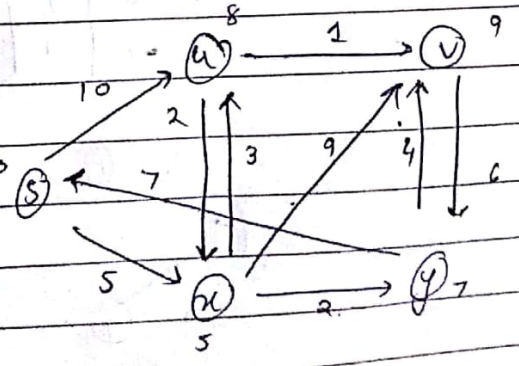
1st → s(0) u(10) v(∞) x(5) y(∞)

2nd → s(0) u(10) v(11) x(5) y(∞)

3rd → s(0) u(8) v(9) x(5) y(7)

4th → s(0) u(8) v(9) x(5) y(7)

final graph:



graph doesn't have -ve cycle.