Name : Shivam Sundriyal

Sec : D

Roll No : 37

PAGE No.
DATE

TUTORIAL - 6

1. A Minimum Spanning Tree (MST) is a subset of edges of a connected weighted undirected graph that connects all the vertices together with the minimum possible total edge weight.

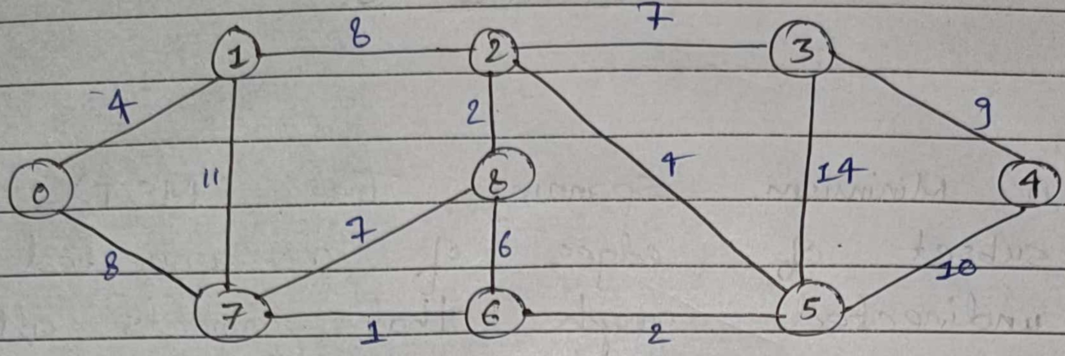
Applications :-

- Network Design
- Telecommunications networks
- Transportation networks
- TV Cable

2.

Algorithm	Time	Space
Kruskal's Algorithm	$O(E \log V)$	$O(E+V)$
Prim's Algorithm	$O(V^2)$ $O(E \log V)$	$O(E+V)$
Djikstra's Algorithm	$O(V^2)$ $O(E \log V)$	$O(V^2)$
Bellman ford Algorithm	$O(VE)$ $O(E)$	$O(V)$

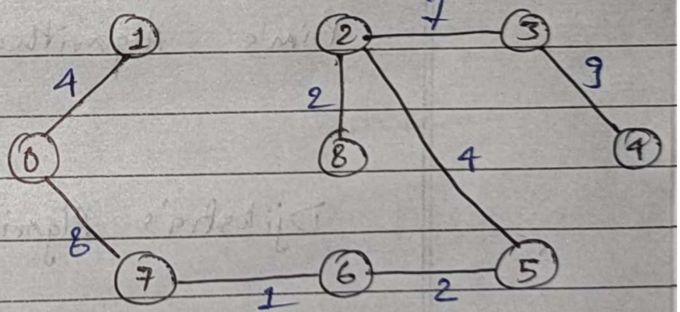
3.



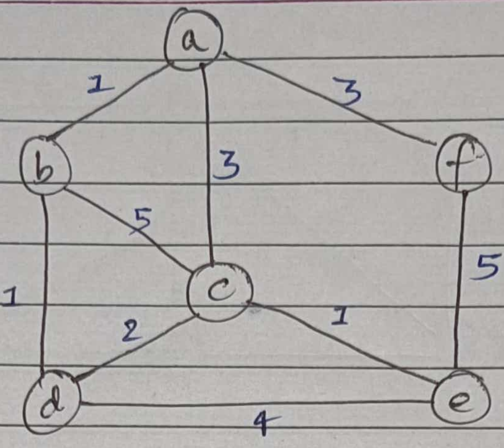
Kruskal's

→ Sort edges in ascending order of weight.
→ Pick an edge with min. weight & push it to result
→ Continue this for V-1 edges until cycle does not come

u	v	wt.
7	6	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	6
3	4	9
5	4	10
1	7	11
3	5	14

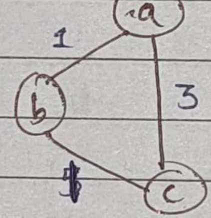


4.



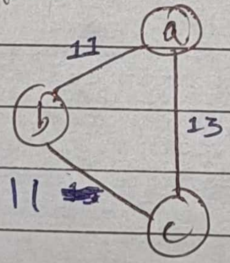
→ If we add the weights of the graph by 10, the shortest path will change

eg: Initially (Consider)



shortest path:
$a \to b \to c$

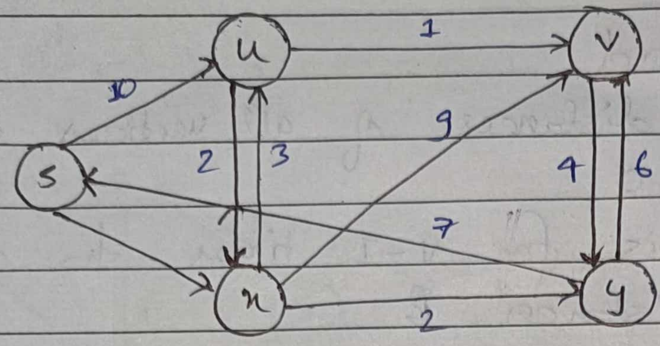
After adding 10



Shortest path
$a \to c$

→ There is no change in the shortest path if we multiply all the weights of the edges by 10.

5.



Dijkstra's

→ Create sptSet which keeps track of vertices.

→ We assign all the vertices with distance infinite. Then we assign distance of source node to 0.

→ While sptSet does not include all the vertices:

(a) Pick a vertex which is not in sptSet & has min. distance

(b) Include it in sptSet

(c) Update distance value of all the adj. vertices of the above vertex using condn

```
if ( dist [v] > dist[u] + graph[u][v])
{
      dist = dist[u] + graph [u][v];
}
```

Node	Shortest dist. from source
s	0
u	8
x	5
v	9
y	7

PAGE No.
DATE

Bellman Ford

→ Initialize distances of all vertices to 0 & assign dist [0] = 0

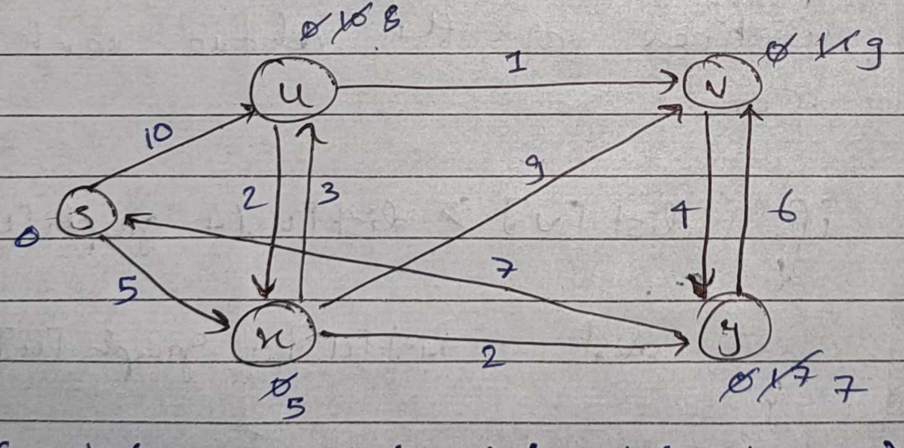
→ Repeat this for v-1 times to calculate shortest distances

```
for each edge u-v
    if (dist[v] > dist[u] + weight[u][v])
    {
        dist[v] = dist[u] + weight[u][v];
    }
```

→ This will report if there is a (-)ve weight

Do this for each edge u-v

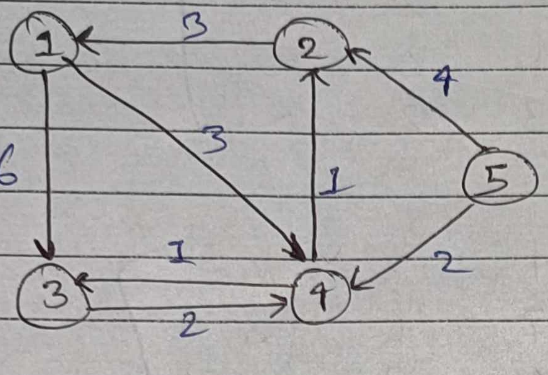
```
if (dist[v] > [dist][u] + weight[u][v])
    cout << "Graph contains negative wt. cycle";
```



(s,u) (s,x) (u,x) (u,v) (v,y) (y,v) (y,s) (x,u) (x,v) (x,y)

Node	Shortest Dist
s	0
u	8
x	5
y	7
v	9

6. Floyd Warshall's



$$G = \begin{bmatrix} 0 & \infty & 6 & 3 & \infty \\ 3 & 0 & \infty & \infty & \infty \\ \infty & \infty & 0 & 2 & \infty \\ \infty & 1 & 1 & 0 & \infty \\ \infty & 4 & \infty & 2 & 0 \end{bmatrix}$$

$$G1 = \begin{bmatrix} 0 & \infty & 6 & 3 & \infty \\ 3 & 0 & 9 & 6 & \infty \\ \infty & \infty & 0 & 2 & \infty \\ \infty & 1 & 1 & 0 & \infty \\ \infty & 4 & \infty & 2 & 0 \end{bmatrix}$$

$$G2 = \begin{bmatrix} 0 & \infty & 6 & 3 & \infty \\ 3 & 0 & 9 & 6 & \infty \\ \infty & \infty & 0 & 2 & \infty \\ 4 & 1 & 1 & 0 & \infty \\ 7 & 4 & 13 & 1 & 0 \end{bmatrix}$$

$$G3 = \begin{bmatrix} 0 & \infty & 6 & 3 & \infty \\ 3 & 0 & 9 & 6 & \infty \\ \infty & \infty & 0 & 2 & \infty \\ 4 & 1 & 1 & 0 & \infty \\ 7 & 4 & 13 & 2 & 0 \end{bmatrix}$$

$$G4 = \begin{bmatrix} 0 & 4 & 4 & 3 & \infty \\ 3 & 0 & 7 & 6 & \infty \\ 6 & 3 & 0 & 2 & \infty \\ 4 & 1 & 1 & 0 & \infty \\ 6 & 3 & 3 & 2 & 0 \end{bmatrix}$$

$$G5 = \begin{bmatrix} 0 & 4 & 4 & 3 & \infty \\ 3 & 0 & 7 & 6 & \infty \\ 6 & 3 & 0 & 2 & \infty \\ 4 & 1 & 1 & 0 & \infty \\ 6 & 3 & 2 & 2 & 0 \end{bmatrix}$$