- Name : Divyansh Gautam
- Univ. roll no. : 2015243
- Section : ML
- Class roll no. : 26
- Subject : Design and Analysis of Algorithms
- Subject code : TCS 505

Tutorial - 6

Q.1 What do you mean by minimum spanning tree? What are the applications of MST?

→ The cost of spanning tree is the sum of weights of all the edges in the tree. There can be many spanning trees. Minimum spanning tree is a spanning tree where the cost is minimum among all the spanning trees. There can be multiple minimum spanning trees also.

Minimum spanning tree has direct application in the design of networks. It is used in algorithms approximating the travelling salesman problem, multi-terminal minimum cut problem and minimum-cost weighted perfect matching. Its other practical applications are:

1) Image Segmentation
2) Handwriting recognition
3) Cluster analysis

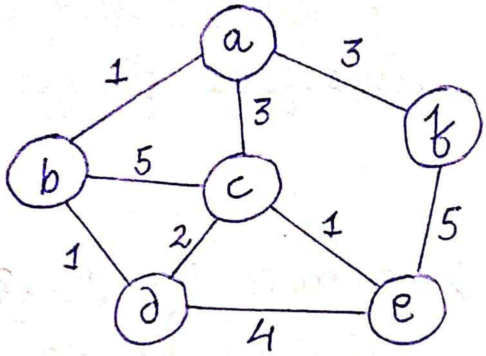
Q.2 Analyse the space and time complexities of Prim's, Kruskal's, Dijkstra's and Bellman Ford's algorithms.

→

Algorithm	Time complexity	Space complexity
Prim's algorithm	$O(V^2)$ $O(E \log V)$ using Fibbonacci heaps	$O(\lvert V \rvert)$
Kruskal's algorithm	$O(E \log E) = O(E \log V)$	$O(\lvert V \rvert)$
Dijkstra's algorithm	$O(E \log V)$ using Priority Queue	$O(\lvert V \rvert + \lvert E \rvert)$
Bellman Ford's algorithm	$O(V \cdot E)$	$O(V)$

Q.3 Apply Kruskal's and Prim's algorithm on graphs given below to compute MST and its weight:

(a)

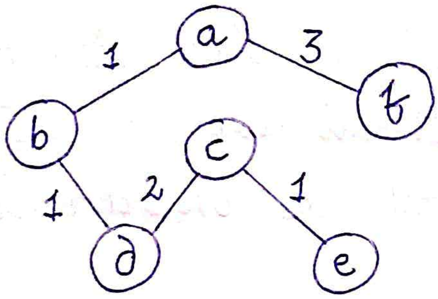


→ Applying Kruskal's algorithm,

u	v	W
a	b	1 ✓
b	d	1 ✓
c	e	1 ✓
c	d	2 ✓
a	c	3 X
a	f	3 ✓
d	e	4
b	c	5
e	f	5

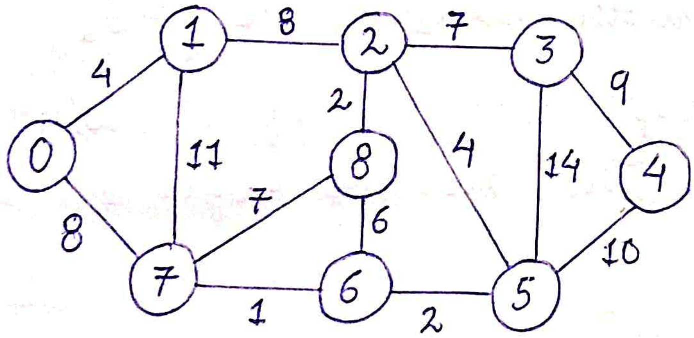
Edges in increasing order of weights

∴ MST



Minimum weight = 8

(b)

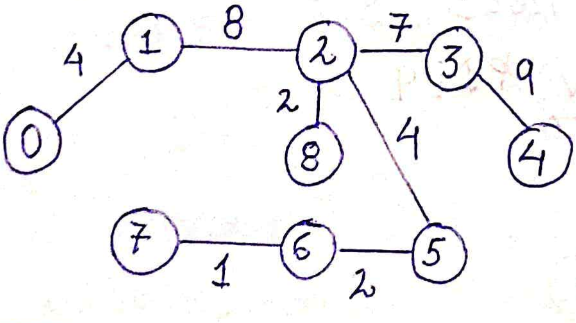


→ Applying Prim's algorithm,

E	W
(0, 1)	7 ✓
(0, 7)	8 X
(1, 7)	11 X
(1, 2)	8 ✓
(2, 3)	7 ✓
(2, 8)	2 ✓
(2, 5)	4 ✓
(8, 7)	7 X
(8, 6)	6 X
(5, 6)	2 ✓
(5, 3)	14 X
(5, 4)	10
(6, 7)	1 ✓

(3, 4)	9 ✓

∴ MST



Minimum weight = 37

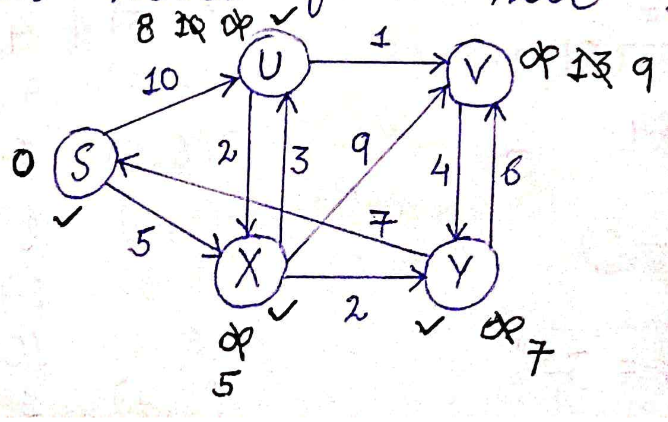
Q.4 Given a directed weighted graph. You are also given the shortest path from a source vertex 'S' to a destination vertex 'T'. Does the shortest path remain same in the modified graph in following cases:

(a) If weight of every edge is increased by 10 units?
(b) If weight of every edge is multiplied by 10 units?

→ (a) The shortest path may change. The reason is, there may be a different no. of edges in different paths from 'S' to 'T'. e.g. Let shortest path be of weight 15 and has 5 edges. Let there be another path with 2 edges & total weight 25. The weight of the shortest path is increased by $5 * 10$ and becomes $15 + 50$.
Weight of other path is increased by $2 * 10$ & becomes $25 + 20$. So, the shortest path changes to the other path with weight as 45.

(b) If we multiply all edge weights by 10, shortest path doesn't change. The no. of edges on a path doesn't matter.

Q.5 Apply Dijkstra's and Bellman's algorithm on the graph given below to compute shortest path to all nodes from node 'S':



→ Applying **Dijkstra's** algorithm,

Node	Shortest distance from source node 'S'
U	8
V	9
X	5
Y	7

Applying **Bellman Ford's** algorithm,

- Initially,

S	U	V	X	Y
0	∞	∞	∞	∞

- Iteration-1

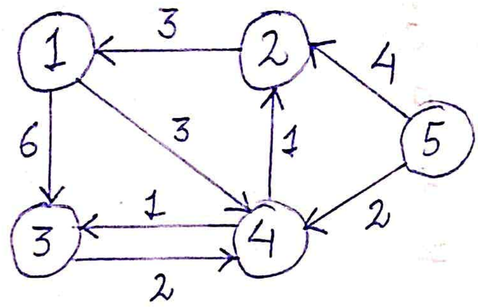
S	U	V	X	Y
0	~~∞~~ ~~10~~ 8	~~∞~~ 11	~~∞~~ 5	~~∞~~ ~~15~~ 7
0	8	11	5	7

- Iteration-2

S	U	V	X	Y
0	8	~~11~~ 9	5	7
0	8	9	5	7

Node	Shortest distance from source node 'S'
U	8
V	9
X	5
Y	7

Q.6 Apply all pair shortest path algorithm - Floyd Warshall's on the given graph and also analyse the time & space complexity of algorithm:



→

$$D_0 = \begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0 & \infty & 6 & 3 & \infty \\ 2 & 3 & 0 & \infty & \infty & \infty \\ 3 & \infty & \infty & 0 & 2 & \infty \\ 4 & \infty & 1 & 1 & 0 & \infty \\ 5 & \infty & 4 & \infty & 2 & 0 \end{array}$$

$$D_1 = \begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0 & \infty & 6 & 3 & \infty \\ 2 & 3 & 0 & 9 & 6 & \infty \\ 3 & \infty & \infty & 0 & 2 & \infty \\ 4 & \infty & 1 & 1 & 0 & \infty \\ 5 & \infty & 4 & \infty & 2 & 0 \end{array}$$

- $A^0[2,3]$ $\quad$ $A^0[2,1] + A^0[1,3]$
  $\infty \; > \; 3 + 6 = 9$
- $A^0[2,4]$ $\quad$ $A^0[2,1] + A^0[1,4]$
  $\infty \; > \; 3 + 3 = 6$
- $A^0[2,5]$ $\quad$ $A^0[2,1] + A^0[1,5]$
  $\infty \; = \; 3 + \infty$
- $A^0[3,2]$ $\quad$ $A^0[3,1] + A^0[1,2]$
  $\infty \; = \; \infty + \infty$
- $A^0[3,4]$ $\quad$ $A^0[3,1] + A^0[1,4]$
  $2 \; < \; \infty + 3$
- $A^0[3,5]$ $\quad$ $A^0[3,1] + A^0[1,5]$
  $\infty \; = \; \infty + \infty$
- $A^0[4,2]$ $\quad$ $A^0[4,1] + A^0[1,2]$
  $1 \; < \; \infty + \infty$
- $A^0[4,3]$ $\quad$ $A^0[4,1] + A^0[1,3]$
  $1 \; < \; \infty + 6$
- $A^0[4,5]$ $\quad$ $A^0[4,1] + A^0[1,5]$
  $\infty \; = \; \infty + \infty$

- $A^0[5,2]$ $\quad$ $A^0[5,1] + A^0[1,2]$
  $4 \; < \; \infty + \infty$
- $A^0[5,3]$ $\quad$ $A^0[5,1] + A^0[1,3]$
  $\infty \; = \; \infty + 6$
- $A^0[5,4]$ $\quad$ $A^0[5,1] + A^0[1,4]$
  $2 \; < \; \infty + 3$

$$D_2 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & \infty & 6 & 3 & \infty \\ 3 & 0 & 9 & 6 & \infty \\ \infty & \infty & 0 & 2 & \infty \\ 4 & 1 & 1 & 0 & \infty \\ 7 & 4 & 13 & 2 & 0 \end{bmatrix} \end{array}$$

$$D_3 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & \infty & 6 & 3 & \infty \\ 3 & 0 & 9 & 6 & \infty \\ \infty & \infty & 0 & 2 & \infty \\ 4 & 1 & 1 & 0 & \infty \\ 7 & 4 & 13 & 2 & 0 \end{bmatrix} \end{array}$$

$$D_4 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 4 & 4 & 3 & \infty \\ 3 & 0 & 7 & 6 & \infty \\ 6 & 3 & 0 & 2 & \infty \\ 4 & 1 & 1 & 0 & \infty \\ 6 & 3 & 3 & 2 & 0 \end{bmatrix} \end{array}$$

$$D_5 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 4 & 4 & 3 & \infty \\ 3 & 0 & 7 & 6 & \infty \\ 6 & 3 & 0 & 2 & \infty \\ 4 & 1 & 1 & 0 & \infty \\ 6 & 3 & 3 & 2 & 0 \end{bmatrix} \end{array}$$

- Time complexity: $O(V^3)$
- Space complexity: $O(V^2)$