Kunal Bandooni
CE
2014413

PAGE No
DATE : / 202

# TUTORIAL-6

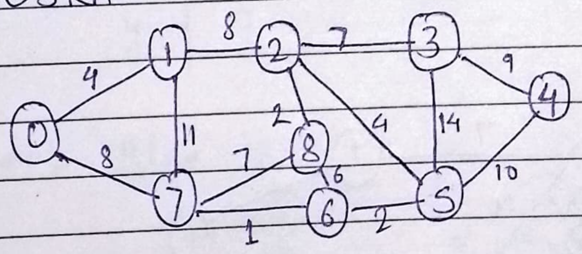
Sol 1. A minimum spanning tree is a spanning tree that has all vertices connected together, without any cycles & with min possible total edge count, i.e., sum of edge weights is minimum.

- Applications of minimum spanning tree
  - Design of network
  - Transport system
  - Minimum distance problems.

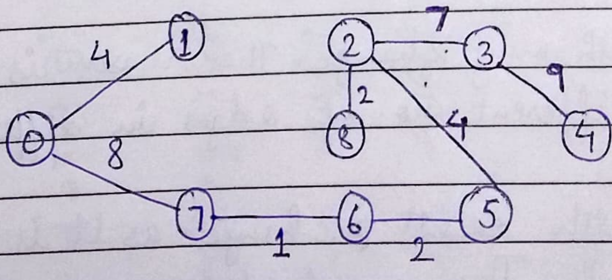
Sol 2.

| Algorithm | Time Complexity | Space Complexity |
|---|---|---|
| Prim's | $O(V^2)$ | $O(1)$ |
| Kruskal's | $O(E \log(V))$ | $O(\lvert E\rvert + \lvert V\rvert)$ |
| Dijkstra's | $O(E \log(V))$ | $O(V^2)$ |
| Bellman Ford | $O(\lvert V\rvert \lvert E\rvert)$ | $O(V)$ |
| Prim's (Adjacency) | $O(E \log(V))$ | $O(V)$ |

Sol 3. • KRUSKAL



| u | v | w | |
|---|---|---|---|
| 7 | 6 | 1 | |
| 6 | 5 | 2 | |
| 2 | 8 | 2 | |
| 2 | 5 | 4 | |
| 0 | 1 | 4 | |
| 6 | 8 | 6 | X |
| 7 | 8 | 7 | X |
| 2 | 3 | 7 | |
| 0 | 7 | 8 | |
| 1 | 2 | 8 | X |
| 3 | 4 | 9 | • |
| 4 | 5 | 10 | X |
| 1 | 7 | 11 | X |
| 3 | 5 | 14 | X |



$\boxed{\text{Total weight} = 37}$

• PRIM'S

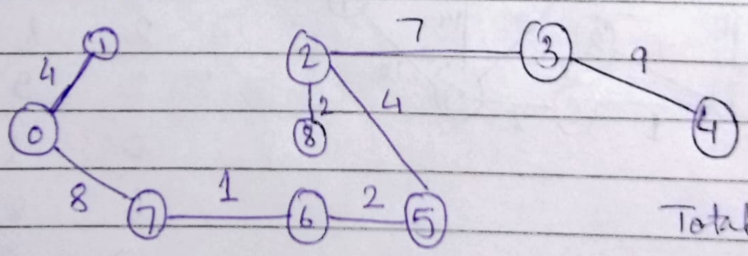
Parent

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

Work Table

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| $\boxed{0}$ | $\boxed{4}$ | ∞ | ∞ | ∞ | ∞ | ∞ | 8 | ∞ |
| | | 8 | ∞ | ∞ | ∞ | ∞ | $\boxed{8}$ | ∞ |
| | | 8 | ∞ | ∞ | ∞ | $\boxed{1}$ | | 7 |
| | | 8 $\boxed{8}$ | ∞ | ∞ | $\boxed{2}$ | | | 6 |
| | | $\boxed{4}$ | 14 | 10 | | | | 6 |
| | | | $\boxed{7}$ | 10 | | | | $\boxed{2}$ |
| | | | | $\boxed{9}$ | | | | |

Final MST

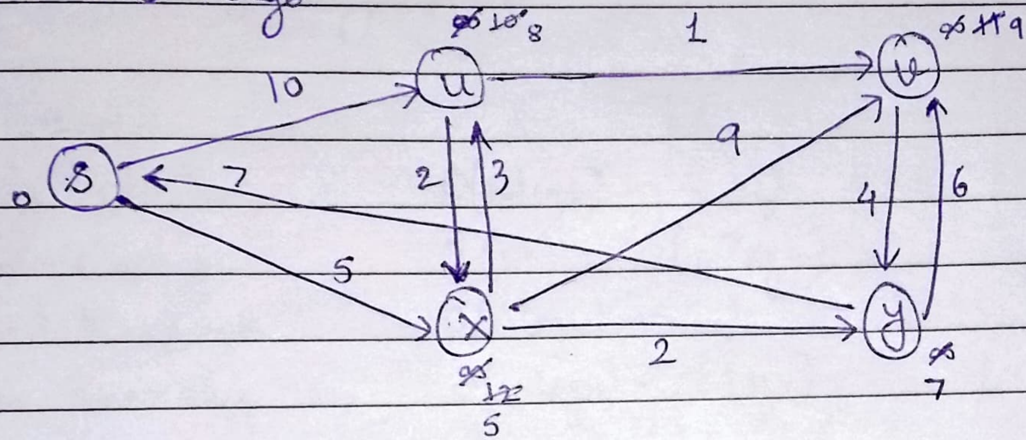



Total weight = 37

Sol4. ① The shortest path may change. The reason is that there may be different no. of edges in different from ~~source~~ (s) ~~to~~ (t).

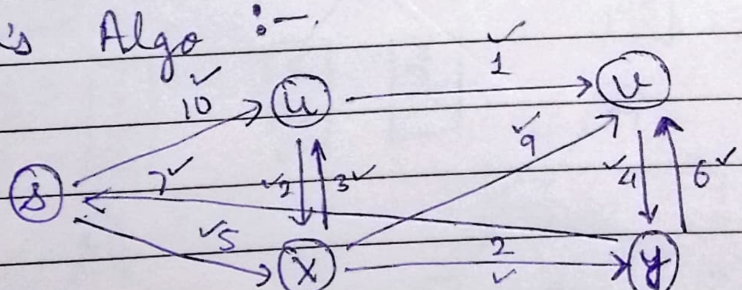
② The shortest path doesn't change as it is merely a scaled graph. The no. of edges on a path doesn't matter here.

## Sols. Dijkstra's Algo :-




| ~~Destination~~ ~~source~~ | distance ~~destination~~ from s |
|---|---|
| u | 8 |
| x | 5 |
| v | 9 |
| y | 7 |

## Bellman's Algo :-




| s | u | v | x | y |
|---|---|---|---|---|
| 0 | ~~∞~~ 10 | ~~∞~~ 11 | ~~∞~~ 5 | ~~∞~~ 7 |
| 0 | ~~10~~ 8 | ~~11~~ 11 | 5 | 7 |
| 0 | 8 | ~~11~~ 9 | 5 | 7 |
| 0 | 8 | 9 | 5 | 7 |

∴ Final Distance

| s | u | v | x | y |
|---|---|---|---|---|
| 0 | 8 | 9 | 5 | 7 |