20

CS5300 Advanced Algorithms Kishor Kumar Andekar

HW #7 700744713 (CRN: 12008)

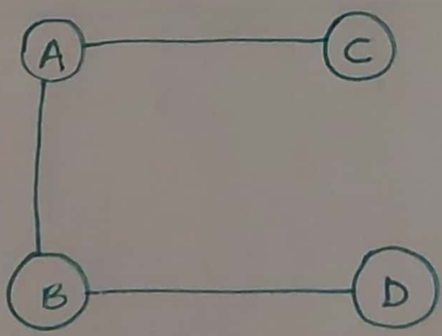
1. Do you think it is possible for a minimum spanning tree to have a cycle ? Justify your Answer.

It is Not possible for a Minimum spanning tree to have a cycle. It is a most economical way of connecting all vertices of a Weighted undirected graph together using the edges of the Graph.

- The Minimum spanning Tree does not have any cycle.
- It has |V| vertices and |V|-1 Edges

Example:

If we have a 4 Vertices and 3 Edges. To get a Loop Number of vertices should be equal to Number of Edges. If Number of vertices and Number of Edges are equal then it is not a Minimum Spanning Tree.

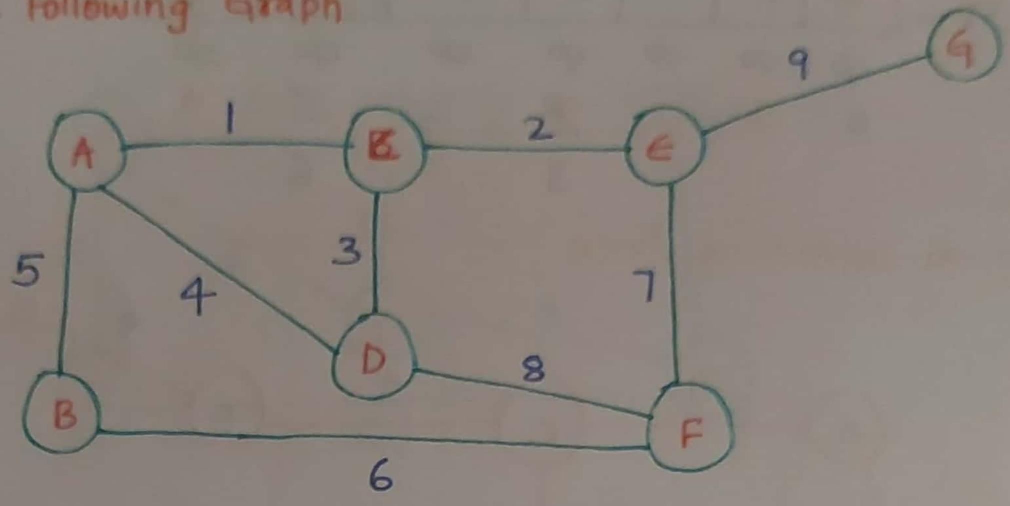



Number of vertices = 4

Number of Edges = 3

2. Use Prim's algorithm to find a minimum spanning tree for the following Graph

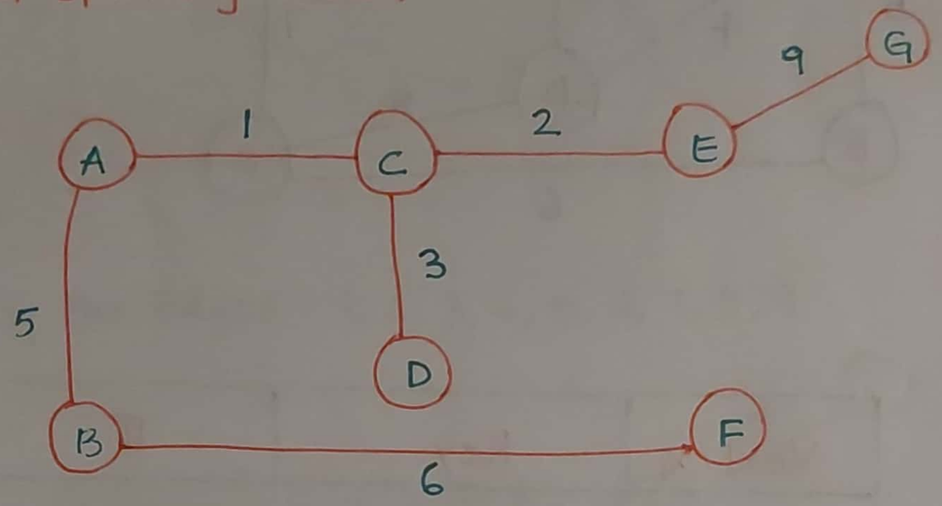
G




| Vertex | Key | π |
| --- | --- | --- |
| A | $\cancel{\infty}$ 0 | Nil |
| B | $\cancel{\infty}$ 5 | ~~Nil~~ A |
| C | $\cancel{\infty}$ 1 | ~~Nil~~ A |
| D | $\cancel{\infty}$ $\cancel{4}$ 3 | ~~Nil~~ ~~A~~ C |
| E | $\cancel{\infty}$ 2 | ~~Nil~~ C |
| F | $\cancel{\infty}$ $\cancel{7}$ 6 | ~~Nil~~ ~~E~~ B |
| G | $\cancel{\infty}$ 9 | ~~Nil~~ E |

| Q | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| | 1 | 5 | 2 | 4 | 3 | 6 | |
| | ~~∞~~ | ~~∞~~ | ~~∞~~ | ~~∞~~ | ~~∞~~ | ~~∞~~ | ~~∞~~ |
| | 0 | 5 | 1 | ~~4~~ | 2 | ~~7~~ | 9 |
| | | | | 3 | | 6 | |

Minimum Spanning Tree :

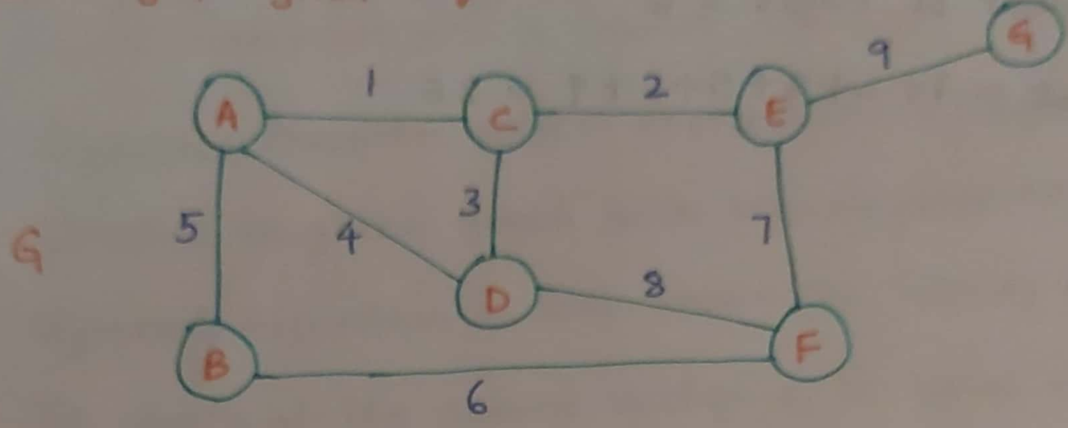



Number of Vertices = 7

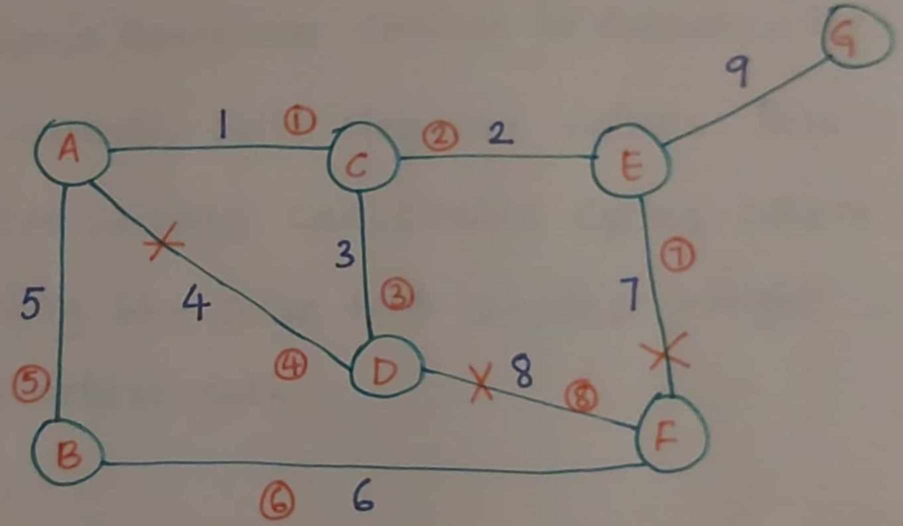
Number of Edges = 6

Cost = 1 + 2 + 3 + 5 + 6 + 9 = 26

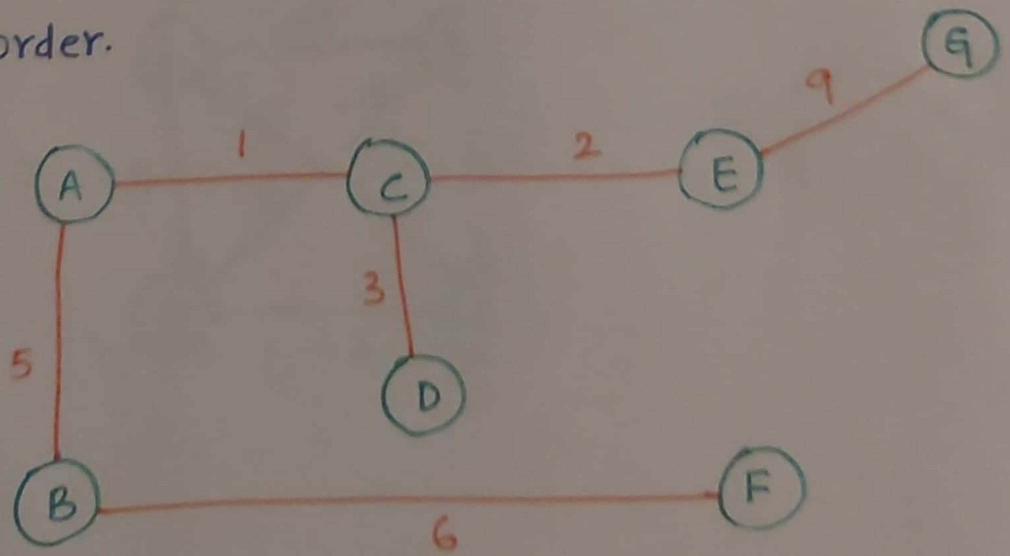
3. Use Kruskal's algorithm to find a minimum spanning tree for the graph given in question 2.




Sorting the Edges : 1, 2, 3, 4, 5, 6, 7, 8, 9




Minimum Spanning Tree after Considering the edges in Sorted order.



Number of Vertices = 7

Number of Edges = 6

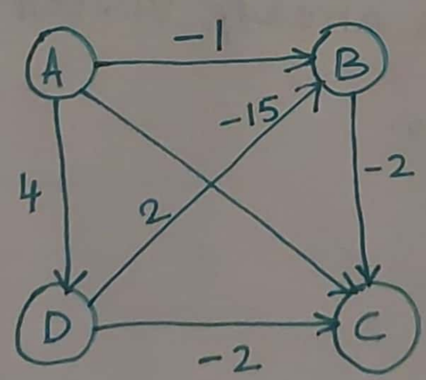
Cost = 1+2+3+5+6+9 = 26

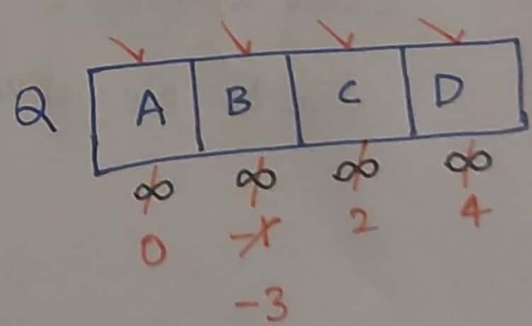
4. Can Dijkstra's algorithm be used to find the shortest paths in a graph with some Negative weights ? If so, prove it; otherwise give a Counter Example.

Dijkstra's Algorithm solves shortest-path problem for directed weighted graph with non-Negative edge Weights. Dijkstra's Algorithm is also Known as Greedy Algorithm. It starts at the source vertex s, it grows a tree T, that eventually spans all vertices reachable from s. Dijkstra's Algorithm confuse to calculate the shortest Path correctly With Negative values. This is because Negative Weights can create cycles Where repeatedly choosing the Edge with Smallest Weight can result in shortest path.

Example :

Let's Implement Dijkstra's Algorithm for below Graph.



| Vertex | d | π |
|---|---|---|
| A | ~~∞~~ 0 | Nil |
| B | ~~∞~~ ~~-1~~ -11 | ~~Nil~~ ~~A~~ D |
| C | ~~∞~~ ~~2~~ -3 | ~~Nil~~ ~~A~~ B |
| D | ~~∞~~ 4 | ~~Nil~~ A |

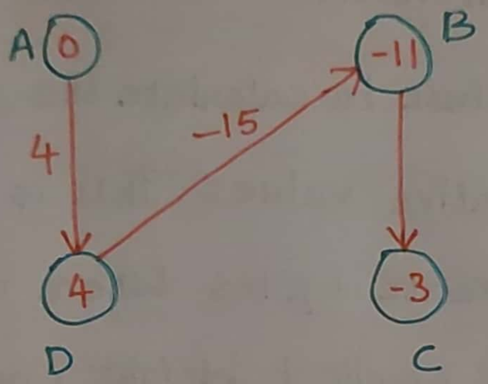
A Adjacent is B, C, D

B Adjacent is C

C has No Adjacent

D has C, B adjacent

Here, At the end we have the shortest path from A as



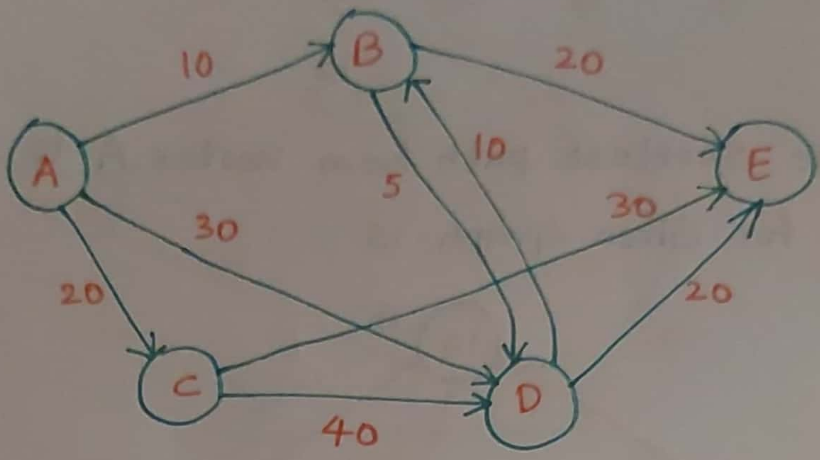

No of Vertices = 4

No of Edges = |V|-1 = 4-1 = 3

Here the problem is for 'C' actual shortest path is from 'B' and it is -13 but it shows '-3' because after Last iteration, 'D'. B is updated but we Cant update 'C' as it is already visited. This is due to Negative values.

5. Use Dijkstra's to find the shortest paths from vertex A to all the other vertices for the graph given below.

G

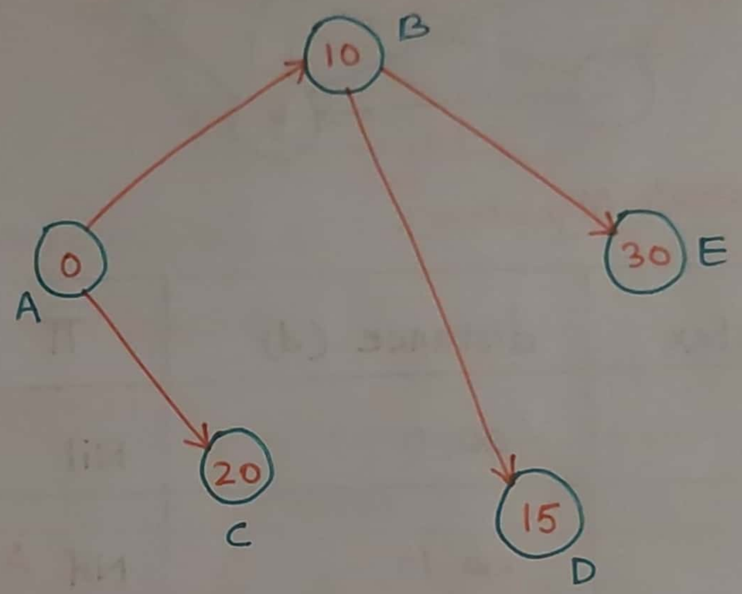



Dijkstra's Algorithm :

| Vertex | distance (d) | $\pi$ |
|---|---|---|
| A | ~~∞~~ 0 | Nil |
| B | ~~∞~~ 10 | ~~Nil~~ A |
| C | ~~∞~~ 20 | ~~Nil~~ A |
| D | ~~∞~~ ~~30~~ 15 | ~~Nil~~ ~~A~~ B |
| E | ~~∞~~ 30 | ~~Nil~~ B |

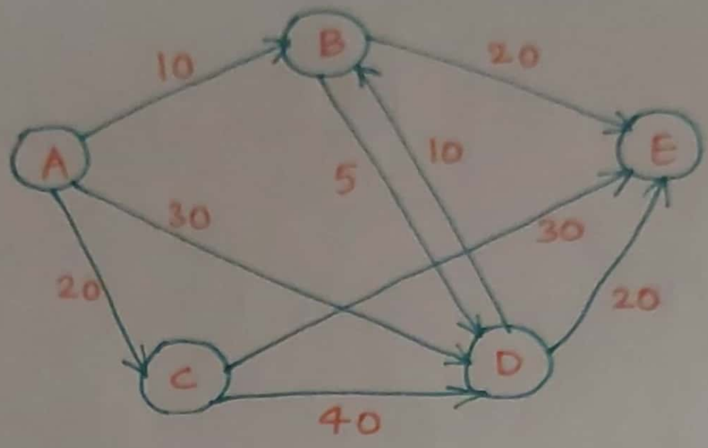
| Q | A | B | C | D | E |
|---|---|---|---|---|---|
| | $\cancel{\infty}$ | $\cancel{\infty}$ | $\cancel{\infty}$ | $\cancel{\infty}$ | $\cancel{\infty}$ |
| | 0 | 10 | 20 | $\cancel{30}$ | 30 |
| | | | | 15 | |

Hence, the Shortest path from vertex A to all other vertices for Given Graph is



6. Use Bellman-Ford Algorithm to find the shortest paths from vertex A to all the other vertices for the graph given in question 5.

G




Given source vertex A

BellMan-Ford Algorithm

$i$ = set of Vertices - 1

| $i$ | Vertex | d | $\pi$ |
|---|---|---|---|
| 1 Vertex 'B', 'C', 'D' update | A | $\cancel{\infty}$ 0 | Nil |
| 2 Vertex 'D', 'E' update | B | $\cancel{\infty}$ 10 | ~~Nil~~ A |
| 3 No update | C | $\cancel{\infty}$ 20 | ~~Nil~~ A |
| 4 No update | D | $\cancel{\infty}$ $\cancel{30}$ 15 | ~~Nil~~ ~~A~~ B |
| 5 STOP | E | $\cancel{\infty}$ 30 | ~~Nil~~ B |

Hence, The shortest path from Vertex A to all other vertices for given Graph is :