Lab report of CSE 206(4)

Course Title: Algorithms Lab
Course Code: CSE 206
Section: PC DA

Submitted to
Dr. Shah Murtaza Rashid Al Masud
Assistant Professor
Dept. of CSE
Green University of Bangladesh

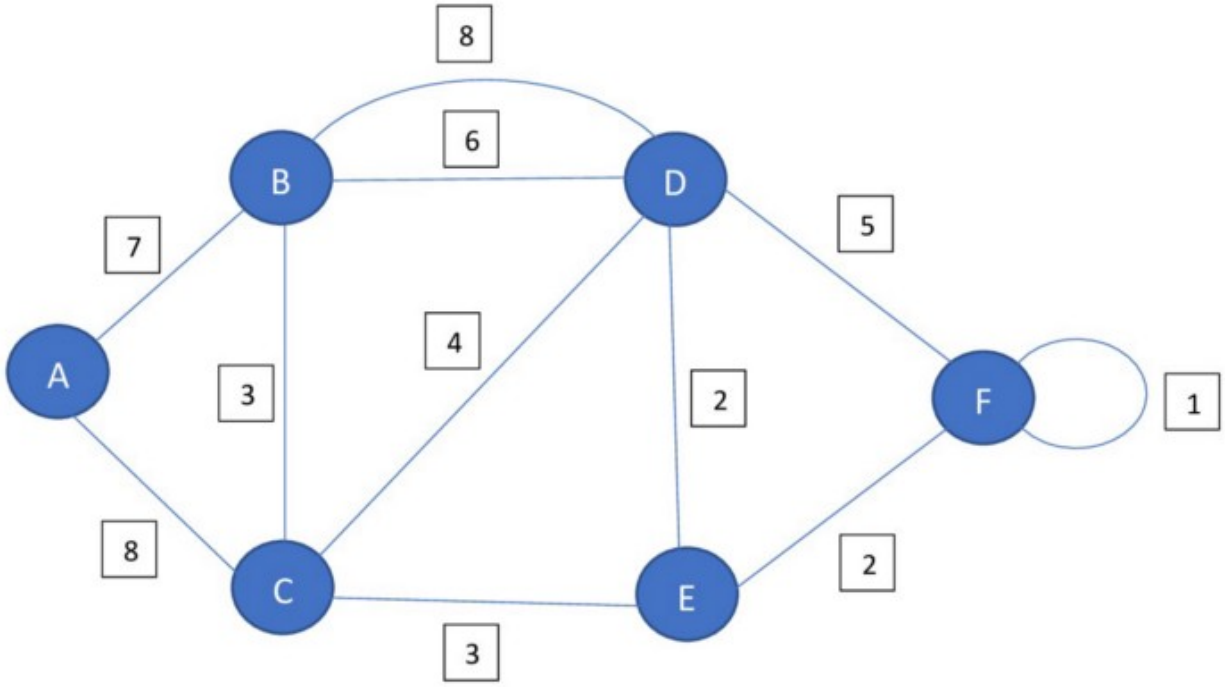
Submitted by:
Mohammad Nazmul Hossain
ID:193902031
Dept. of CSE

Prim's Algorithm

Prim's is a greedy algorithm that uses the shortest pathfinding algorithm and prims help us finds a minimum spanning tree (MST) for a weighted undirected graph. It is also known as Jarnik's algorithm. It was developed by a Czech mathematician named Vojtech Jarnik in 1930. Other known algorithms for this problem are Kruskal's algorithm and Boruvka's algorithm. So, the basic idea is that we must choose a vertex (one can choose any vertex), and we must grow the tree only by one edge at a time. And we must choose the minimum weighted edge and transfer that edge to the tree. So, we must repeat this same process until we have all the vertices in the tree.

★ Algorithm

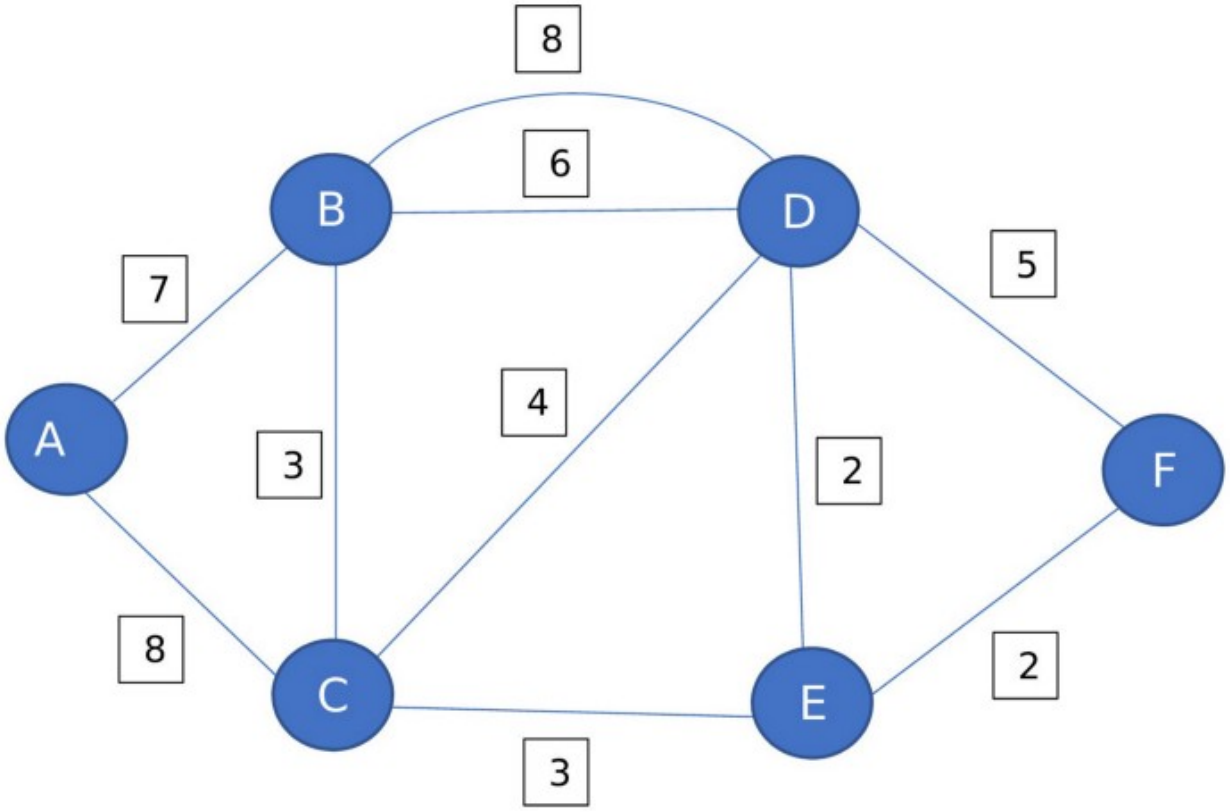
Now that we are going to learn about Prim's algorithm, now let's look at an example to understand its working.



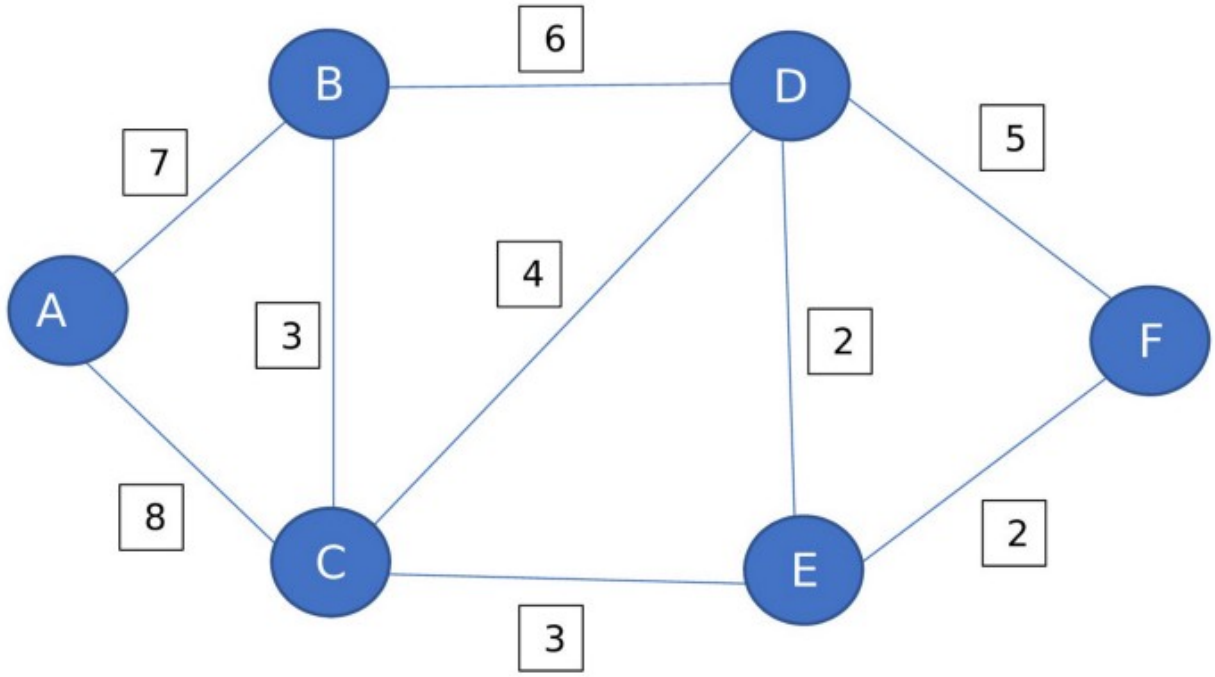
An undirected weighted graph

Now, this is an undirected weighted graph and we have to find its minimum spanning tree using Prim's Algorithm.

Step 1: First we have to remove those edges which are either parallel or are causing a loop(if any). So if you see closely, you will find that vertex F has an edge having weight 1 which is forming a loop, so we have to remove it. So now the graph looks……

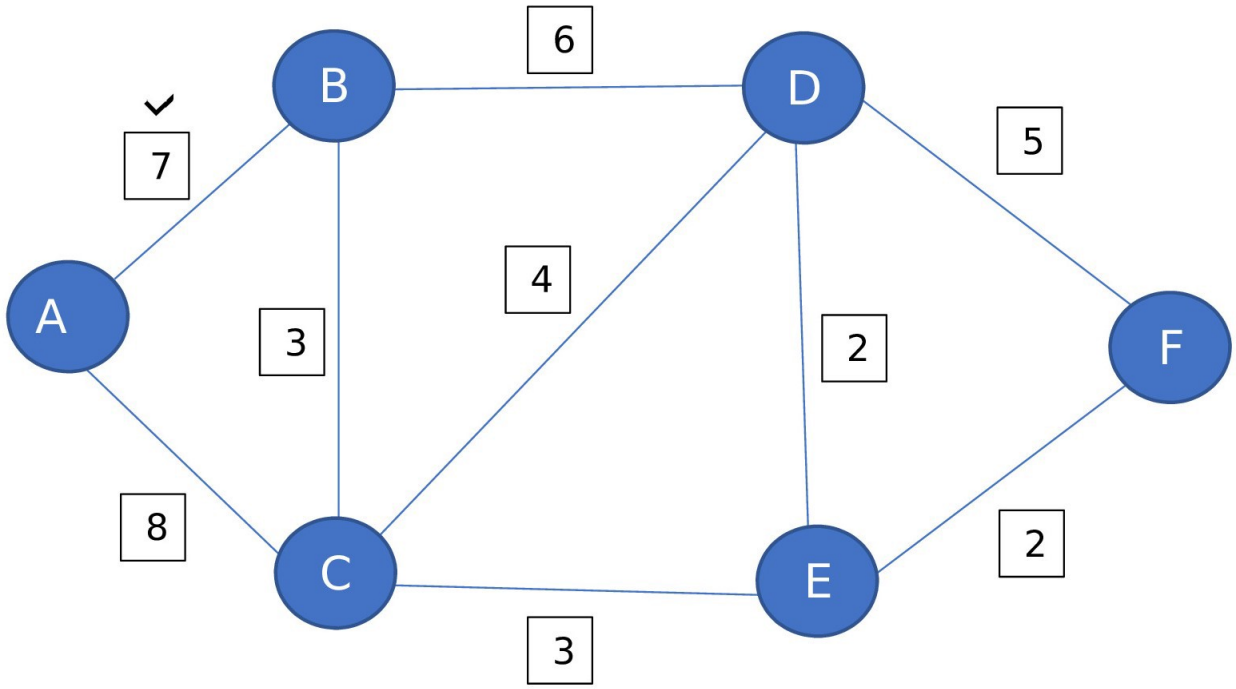


Step 2: Now we have to remove the parallel edges(if any). The two edges between vertices B and D are forming parallel edges. But how will we know which edge to delete? So here we will delete the edge having a higher weight cause we are trying to make an MST here so the weight of the edge should be minimum so that's why we are choosing an edge with having the weight of 6 and removing the edge with having a weight of 8. Now the given problem becomes something like this.

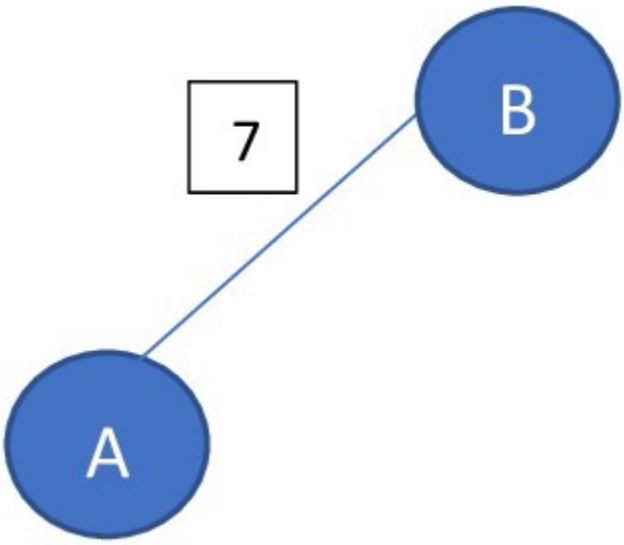


Step 3: Now we can start to form our minimum spanning tree. Choose any one vertex from the above graph(In this example we have selected vertex A). So initially our minimum spanning tree looks like the image given below.

Step 4: Now we will choose that edge that is connected to A and having the minimum weight. So we will choose vertex B cause its weight of 7 is less than the weight of vertex C which is 8. So now the current scenario looks like this.



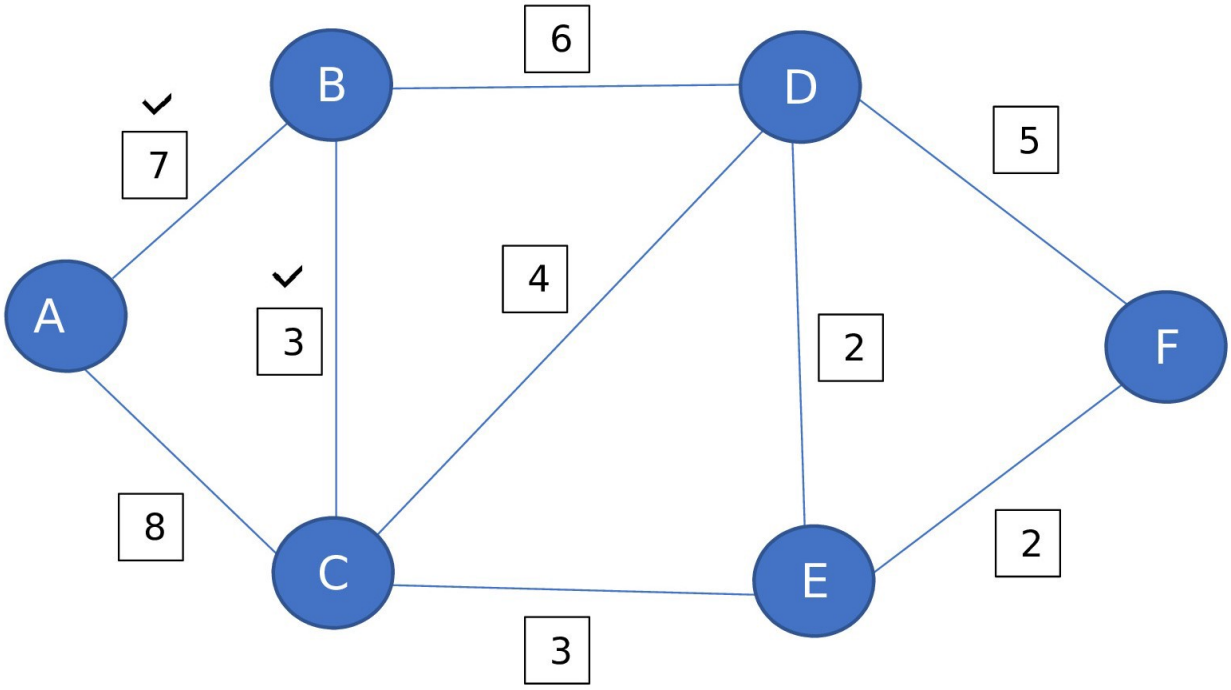
Current Graph

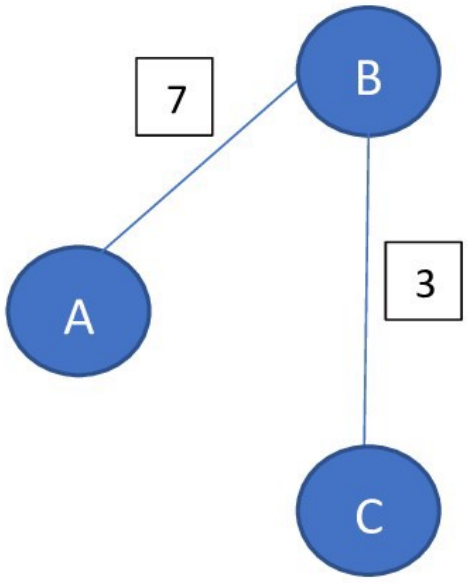


Current Minimum Spanning Tree

Step 5: Now we have to see the least weight from vertex B. But we should not forget that we still have to consider weights of previous edges also like while comparing weights 6 and 8 from vertex B, we also have to compare them with

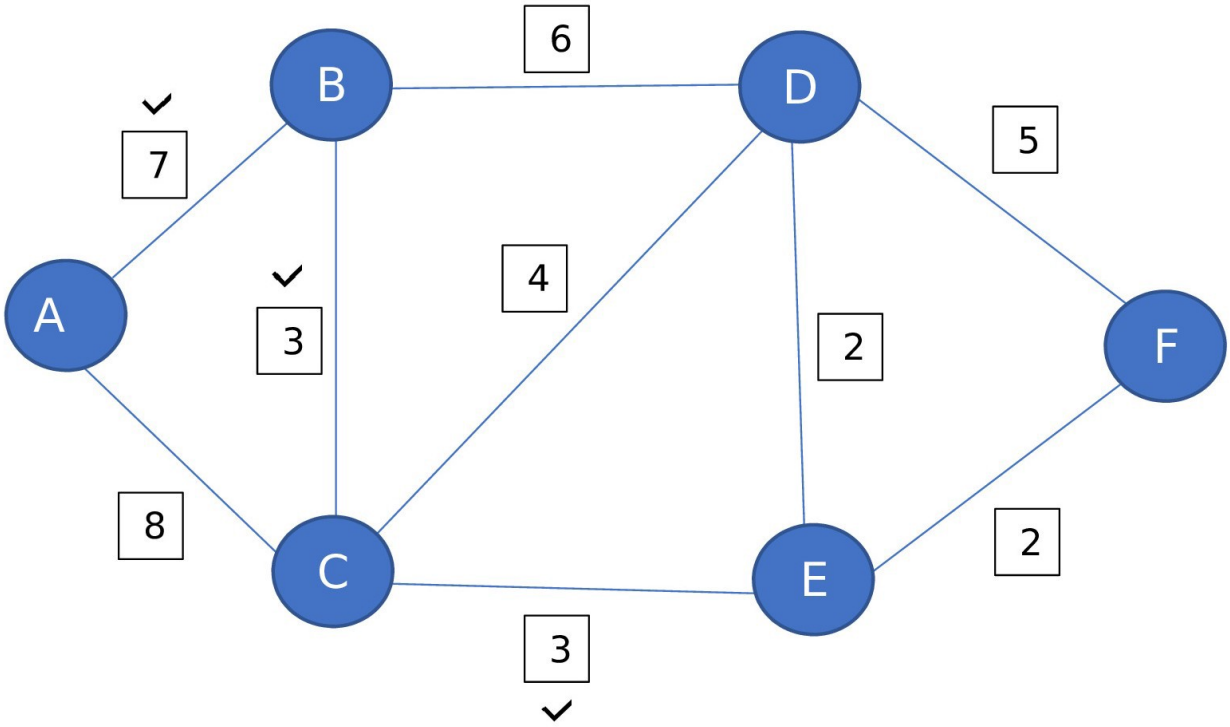
weight 8 from vertex A. So here we will select the minimum which is 3. So now the situation will be…..

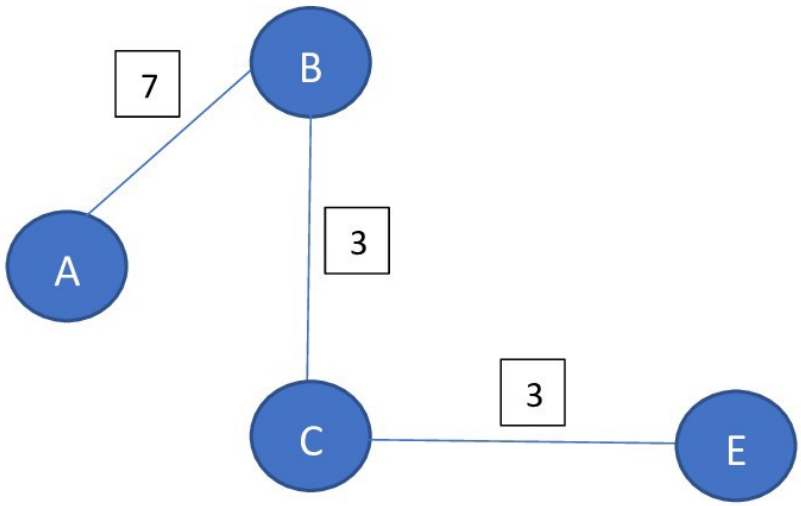




Current Minimum Spanning Tree\

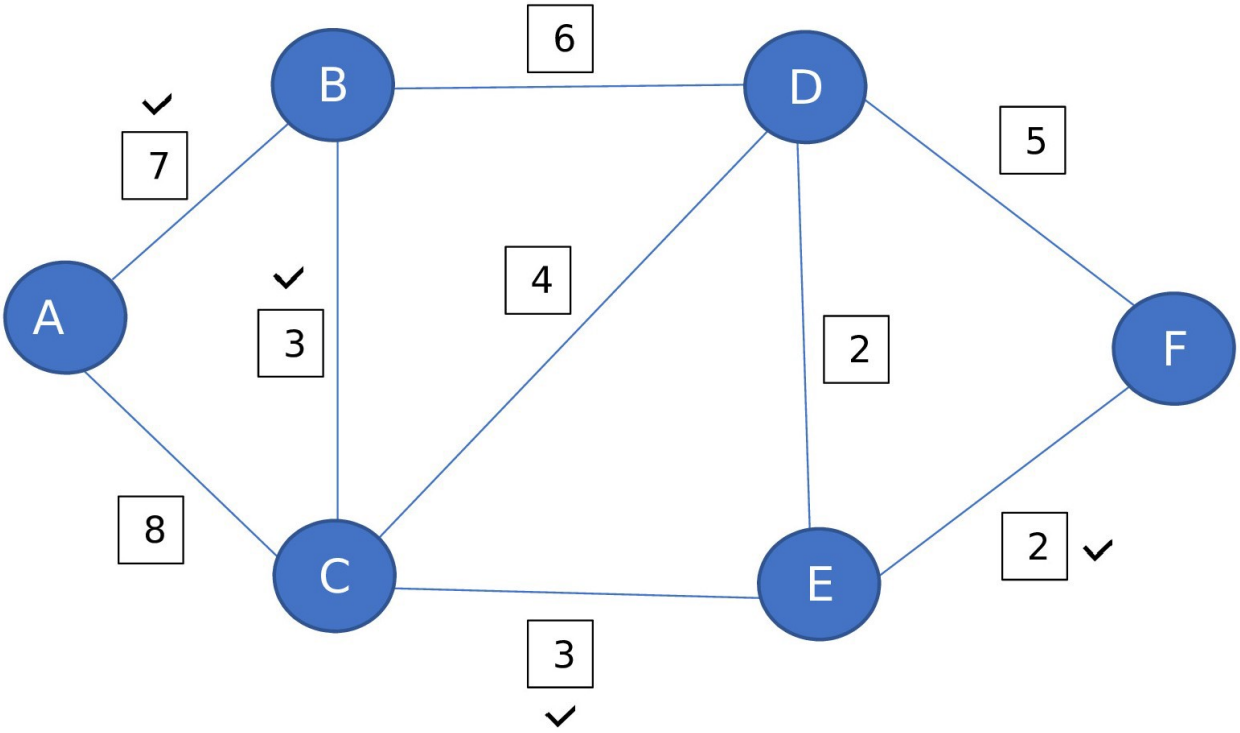
Step 6: Similarly repeating the same procedure again we will get weight 3 as a minimum from vertex C.

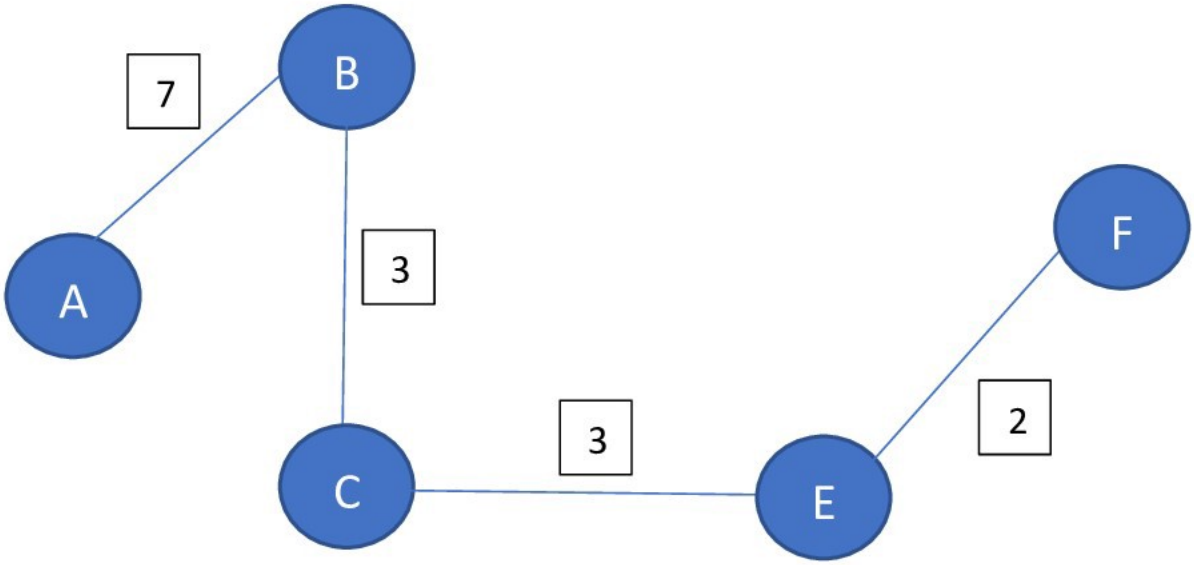




Current Minimum Spanning Tree

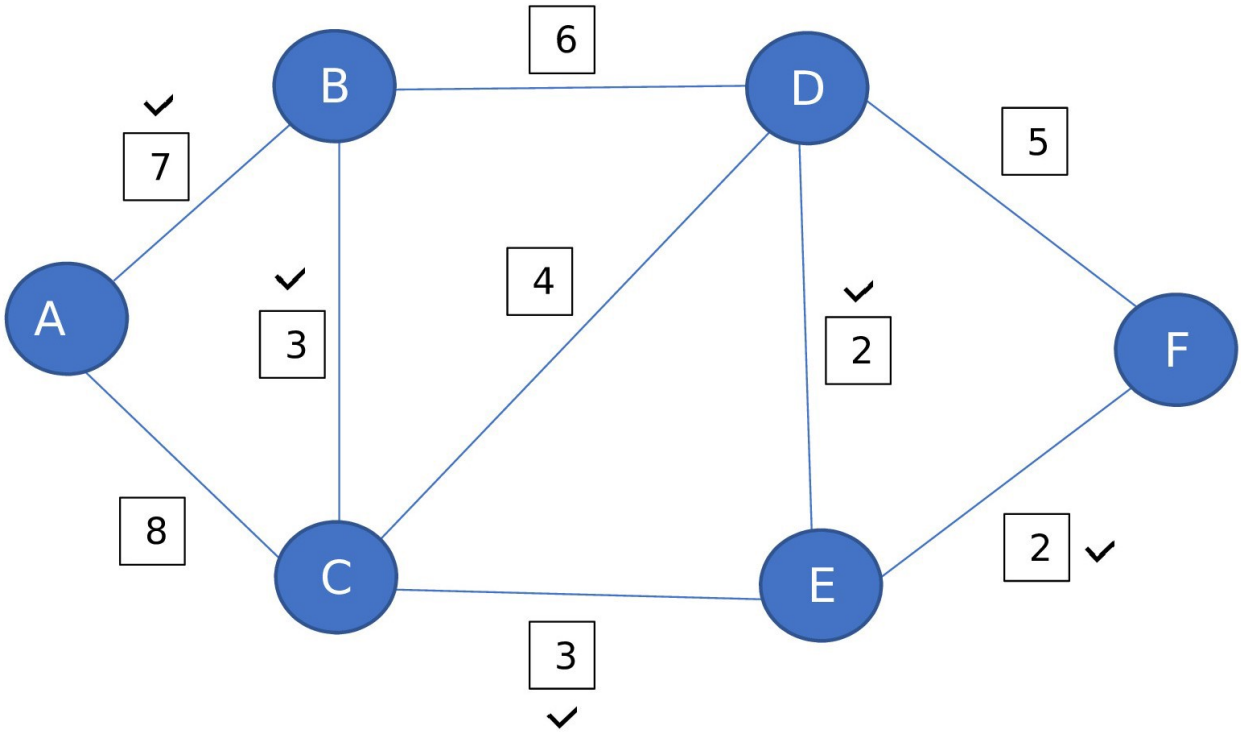
Step 7: Repeating the last step again, we will have 2 as a minimum. But we can see here that there are two edges having a weight of 2, so which one should we select? Actually, we can select either of them(In this example, we are selecting weight between E and F vertices. Now the scenario is……

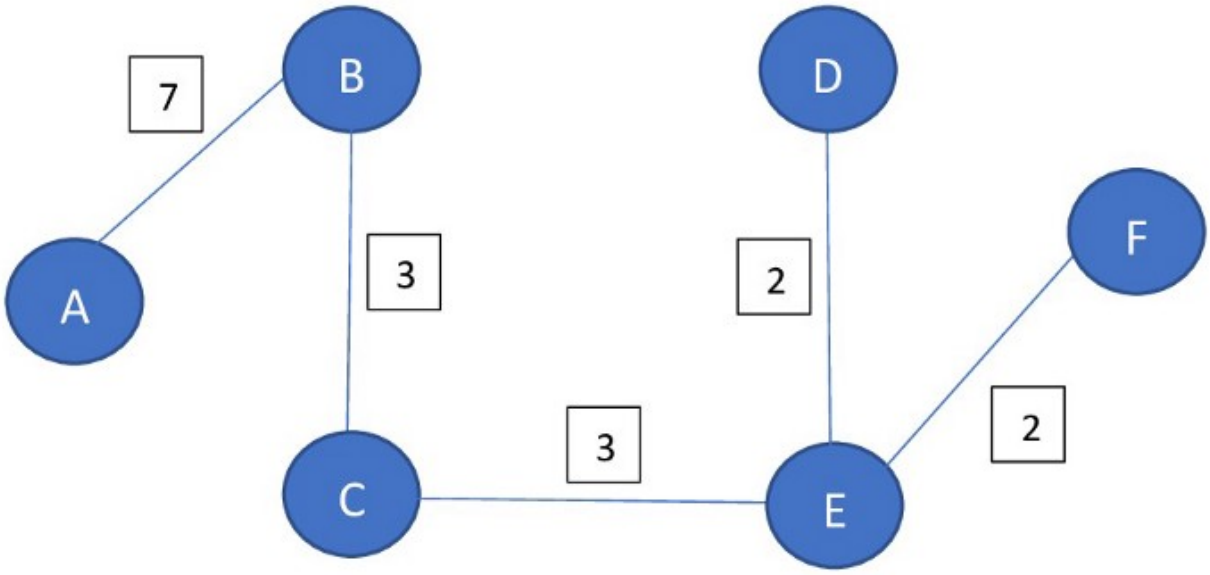




Current Minimum Spanning Tree

Step 8: As we can see that only one vertex remains now which is D so we will select the weight 2 between vertices D and E. So it will look like this…..





Final Minimum Spanning Tree

Now finally, we have covered all vertices and we got our ***minimum spanning tree(MST)*** and this will be the gool.