Assignment No – 07 (Group C)

Problem statement:

You have a business with several offices; you want to lease phone lines to connect them up with each other; and the phone company charges different amounts of money to connect different pairs of cities. You want a set of lines that connects all your offices with a minimum total cost. Solve the problem by suggesting appropriate data structures.

Pre-requisite

- Knowledge of C++ programming
- Knowledge of Minimum Spanning Tree

Objective

- Understanding Minimum Spanning Tree

Input

- Number of vertices ,edges and weight for the edge.

Output

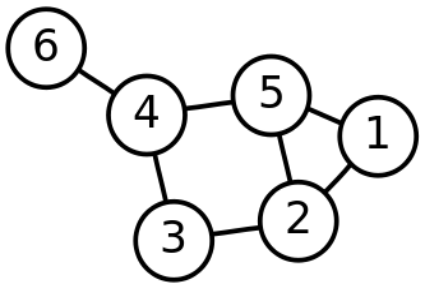
- Minimum Spanning Tree.

Software and Hardware requirements:-

1. **Operating system:** Linux- Ubuntu 16.04 to 17.10, or Windows 7 to 10,
2. **RAM-** 2GB RAM (4GB preferable)
3. C++ / gcc compiler- / 64 bit Fedora, eclipse IDE

Theory-

A graph G is an ordered pair (V, E) where V is the set of vertices and E is the set of edges. Each edge is associated with an unordered pair (vi , vj). The vertices Vi & vj are called the end vertices or the terminal vertices of the edge Eij.



Several Offices in the statement represent vertices of a graph and the lease lines connecting them represent the edges. The objective is to connect all offices by the lease lines with minimum cost. This is the problem of creating Minimum Spanning Tree.

There are two ways to create MST:

1) Prim's Algorithm
2) Kruskal's Algorithm

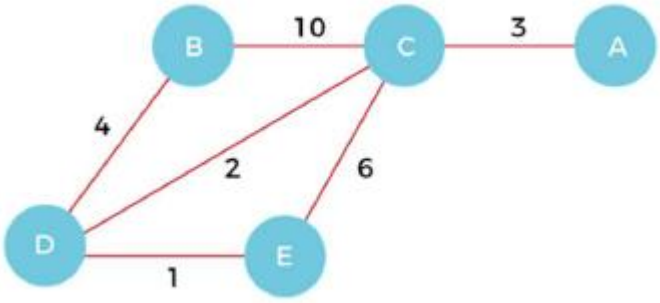
1) Prim's Algorithm:

Step 1: Declare an array visited[] to store the visited vertices which is initially empty
Step 2: Choose any arbitrary vertex say S, as a starting vertex and add it to the visited array.
Step 3: Check whether the adjacent vertices of the last visited vertex are present in the visited[] array or not.
Step 4: If the vertices are not in the visited[] array, compare the cost of edges and add the least cost edge to the MST.
Step 5: The adjacent unvisited vertex with the least cost edge is added into the visited[] array and the least cost edge is added to the minimum spanning tree.
Step 6: Steps 2 and 4 are repeated for all the unvisited vertices in the graph to obtain the full minimum spanning tree output for the given graph.

Step 7: Calculate the cost of the minimum spanning tree obtained.

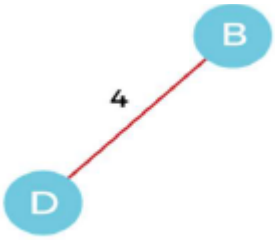
Prim's Algorithm Example:-

Now, let's see the working of prim's algorithm using an example
Suppose, a weighted graph is –

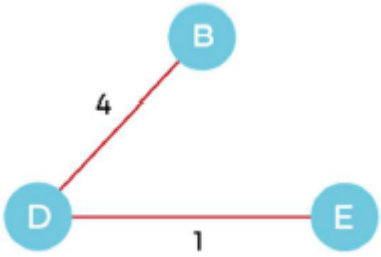


Step 1 - First, we have to choose a vertex from the above graph. Let's choose B.

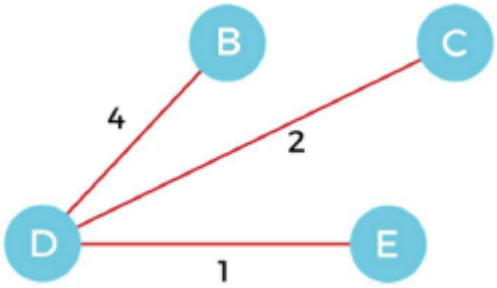
Step 2 - Now, we have to choose and add the shortest edge which is connected to vertex B. There are two edges from vertex B that are B to C with weight 10 and edge B to D with weight 4. Among the edges, the edge BD has the minimum weight. So, add it to the MST.



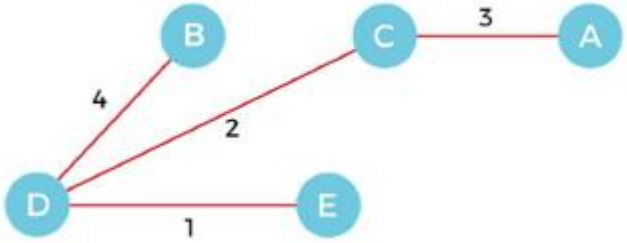
Step 3 - Now, again, choose the edge with the minimum weight among all the edges which are adjacent to vertices B and D. In this case, the edges DE will get added to MST.



Step 4 - Now, choose select the edge with the minimum weight among all the edges which are adjacent to vertices B and D and E .Edge with minimum weight is CD.Add it to the MST.



Step 5 – Similarly we will try to add edge CE but we cannot add the edge CE as it would create a cycle to the graph. So, we will skip it and choose the edge CA which is next minimum weight edge and add it to the MST.

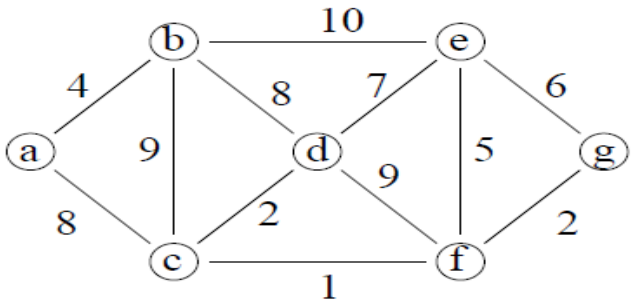


Since all vertices of Graph are now included in MST, the minimum spanning tree of the given graph is produced. The cost of the MST is given below -

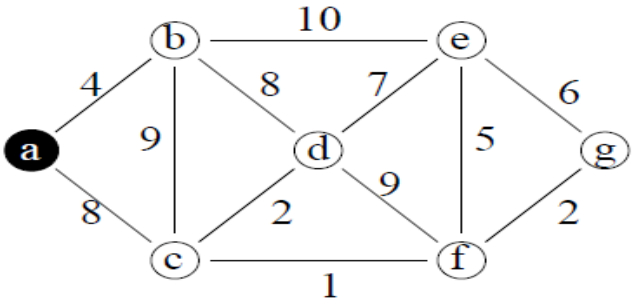
Cost of MST = 4 + 2 + 1 + 3 = 10 units.

Now, let's see the working of prim's algorithm using an example

Suppose, a weighted graph is –



Connected graph

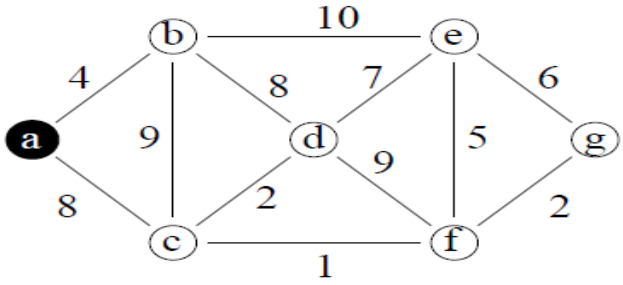


Step 0

S={a}

V \ S = {b,c,d,e,f,g}

lightest edge = {a,b}



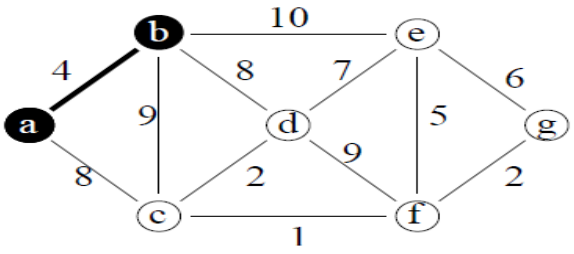
Step 1.1 before

S={a}

V \ S = {b,c,d,e,f,g}

A={}

lightest edge = {a,b}



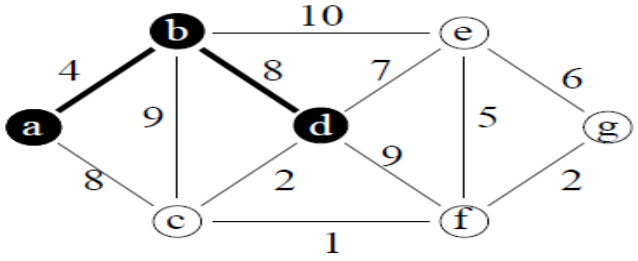
Step 1.1 after

S={a,b}

V \ S = {c,d,e,f,g}

A={{a,b}}

lightest edge = {b,d}, {a,c}



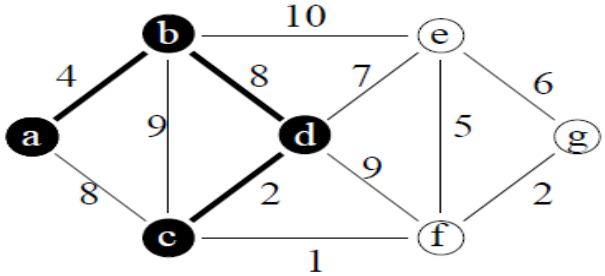
Step 1.2 after

S={a,b,d}

V \ S = {c,e,f,g}

A={{a,b},{b,d}}

lightest edge = {d,c}



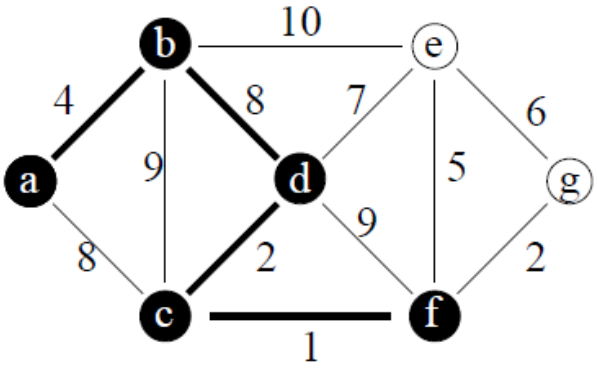
Step 1.3 after

S={a,b,c,d}

V \ S = {e,f,g}

A={{a,b},{b,d},{c,d}}

lightest edge = {c,f}



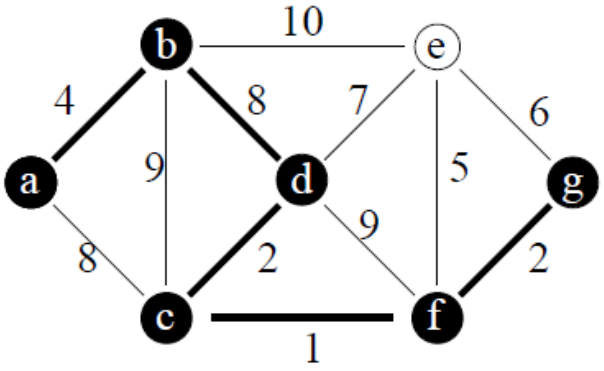
Step 1.4 after

S={a,b,c,d,f}

V \ S = {e,g}

A={{a,b},{b,d},{c,d},{c,f}}

lightest edge = {f,g}



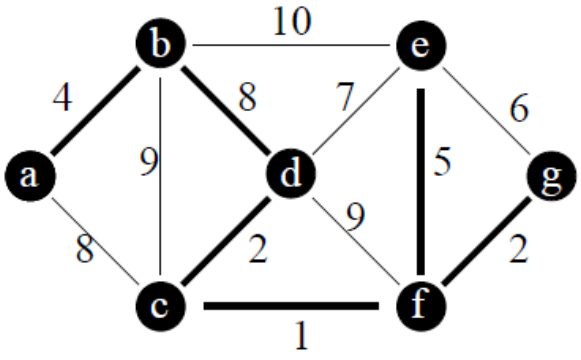
Step 1.5 after

S={a,b,c,d,f,g}

V \ S = {e}

A={{a,b},{b,d},{c,d},{c,f},
{f,g}}

lightest edge = {f,e}



Step 1.6 after

S={a,b,c,d,e,f,g}

V \ S = {}

A={{a,b},{b,d},{c,d},{c,f},
{f,g},{f,e}}

MST completed

Cost of MST = 4 + 8+ 2 +1+ 2+5 = 22 units.

Time Complexity

Prim's algorithm can be simply implemented by using the adjacency matrix or adjacency list graph representation, and to add the edge with the minimum weight requires the linearly searching of an array of weights. It requires O(|V|2) running time. It can be improved further by using the implementation of heap to find the minimum weight edges in the inner loop of the algorithm.

Conclusion:

This program gives us the knowledge of Minimum Spanning Tree

Sample Oral Questions:

1. What is Spanning Tree?
2. What is Minimum Spanning Tree?
3. What is Prim‟s Algorithm?
4. What is Kruskal‟s Algorithm?
5. What are Applications of Minimum Spanning Tree?
6. How to decide whether to select Kruskal‟s algorithm or Prim‟s algorithm
7. What is shortest path algorithm?
8. What is minimum spanning tree of a graph?
9. How to calculate shortest path of graph?
10. Explain Following with example:
    a) Strongly connected components
    b) Spanning Tree
11. What is Minimum Spanning Tree? How is it different than the Shortest path Sequence of the Graph? Justify your answer with example.
12. Discuss about time complexities of prim's and Kruskal's algorithms to find Minimum Spanning Tree.