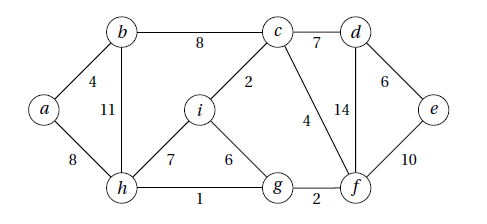
**LAB-7**

**AIM:**

Write a program to

1. Compute the MST of the graph shown in the figure using Prim’s algorithm and Kruskal’s Algorithm.
2. Compute the shortest path using Dijkstra’s Algorithm considering ‘a’ as source vertex to all the vertex.

Graph:



**EXPERIMENT:**

Prim’s Algorithm:

Prim's algorithm is a [minimum spanning tree](https://www.programiz.com/dsa/spanning-tree-and-minimum-spanning-tree#minimum-spanning) algorithm that takes a graph as input and finds the subset of the edges of that graph which

* form a tree that includes every vertex.
* has the minimum sum of weights among all the trees that can be formed from the graph

The steps for implementing Prim's algorithm are as follows:

1. Initialize the minimum spanning tree with a vertex chosen at random.
2. Find all the edges that connect the tree to new vertices, find the minimum and add it to the tree.
3. Keep repeating step 2 until we get a minimum spanning tree

Kruskal’s Algorithm:

Kruskal's algorithm is a [minimum spanning tree](https://www.programiz.com/dsa/spanning-tree-and-minimum-spanning-tree#minimum-spanning) algorithm that takes a graph as input and finds the subset of the edges of that graph which

* form a tree that includes every vertex.
* has the minimum sum of weights among all the trees that can be formed from the graph

The steps for implementing Kruskal's algorithm are as follows:

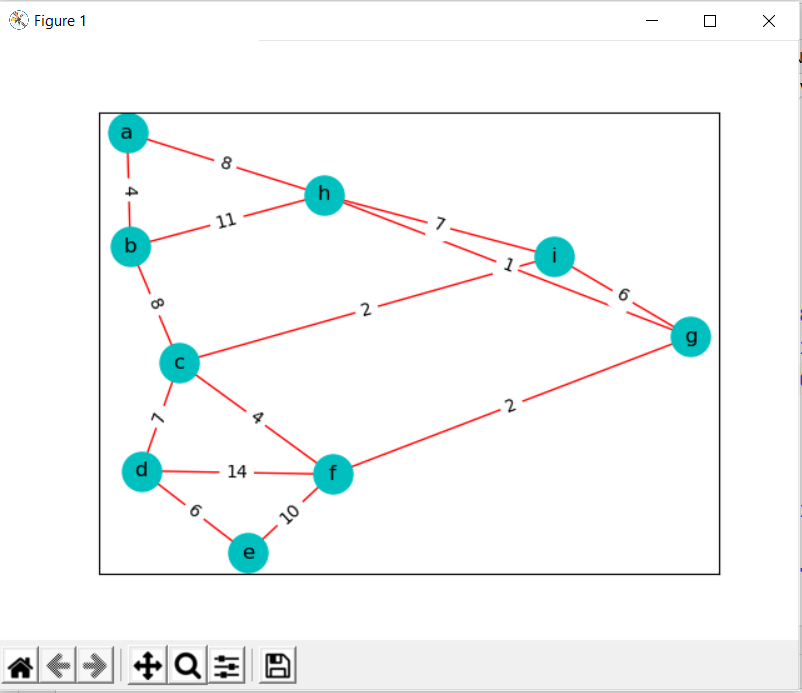
1. Sort all the edges from low weight to high.
2. Take the edge with the lowest weight and add it to the spanning tree. If adding the edge created a cycle, then reject this edge.
3. Keep adding edges until we reach all vertices.

Dijkstra’s Algorithm:

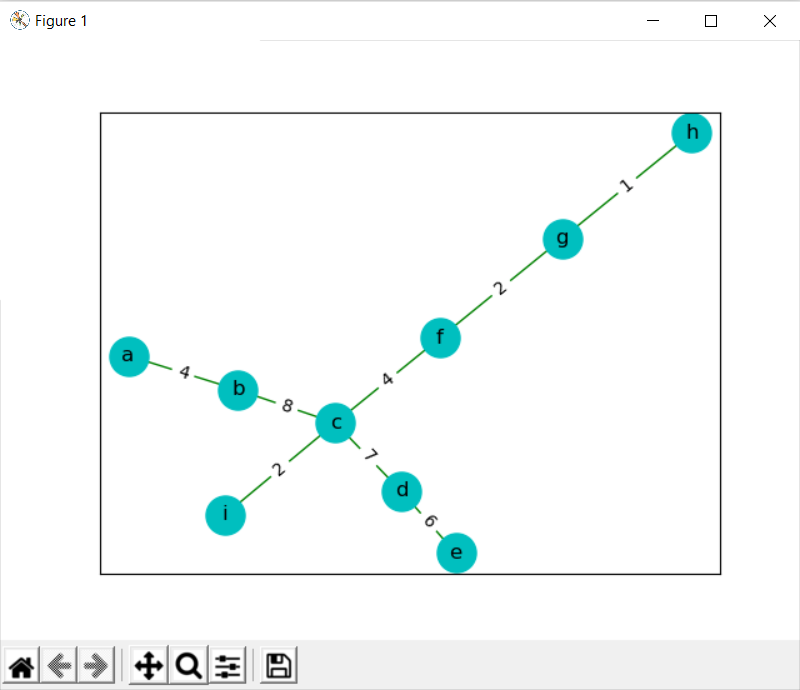
Dijkstra's algorithm allows us to find the shortest path between any two vertices of a graph. It differs from the minimum spanning tree because the shortest distance between two vertices might not include all the vertices of the graph. The algorithm uses a greedy approach in the sense that we find the next best solution hoping that the end result is the best solution for the whole problem.

**OUTPUT:**

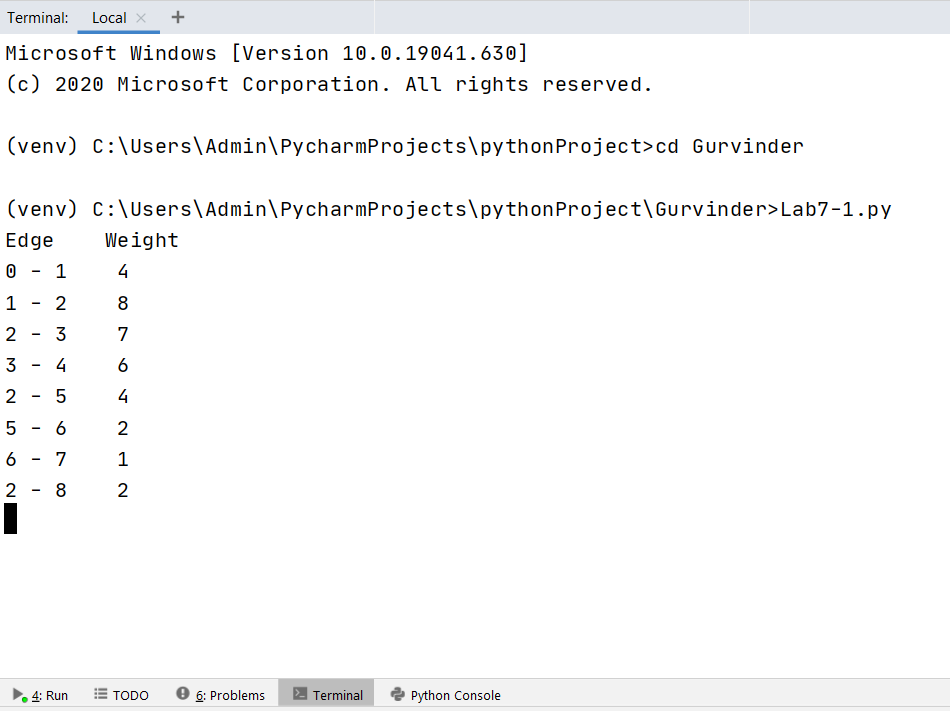
Graph Visualization:



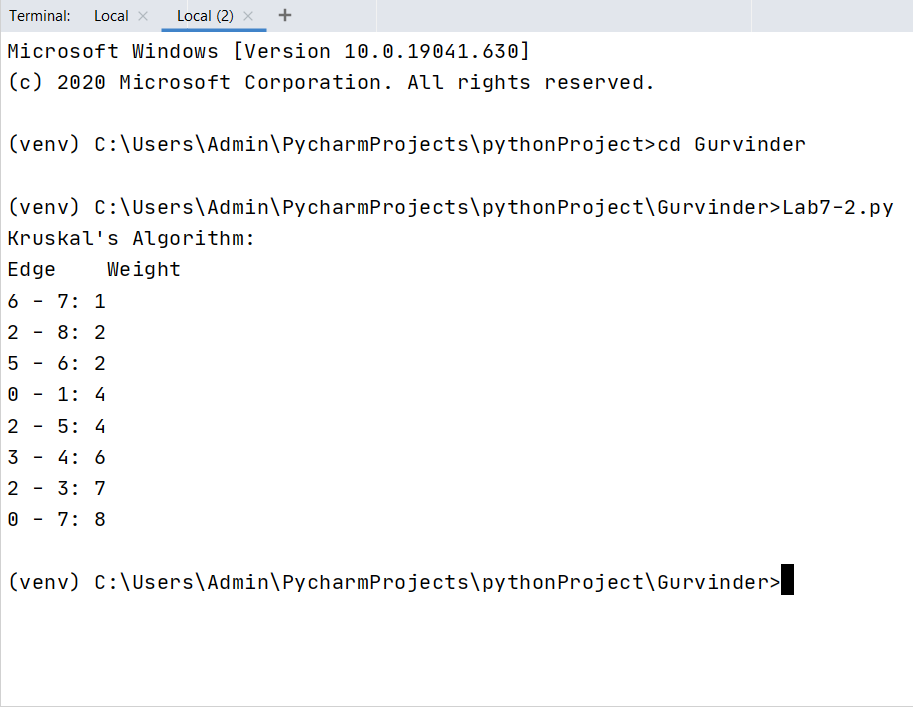
Minimum Spanning Tree:



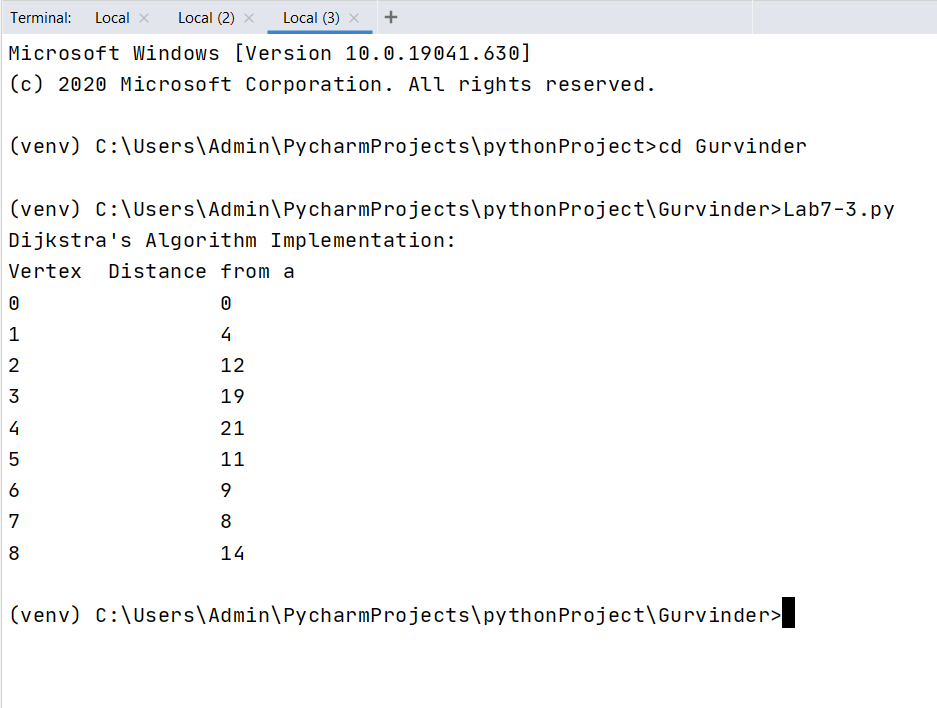
Prim’s Algorithm:



Kruskal’s Algorithm:



Dijkstra’s Algorithm:



**CONCLUSION:**

Thus, Prim’s Algorithm, Kruskal’s Algorithm and Dijkstra’s Algorithm was performed on the given graph successfully.