EXPERIMENT 3

Objective: Implementation of Kruskal's algorithm to find Minimum Spanning Tree (MST).

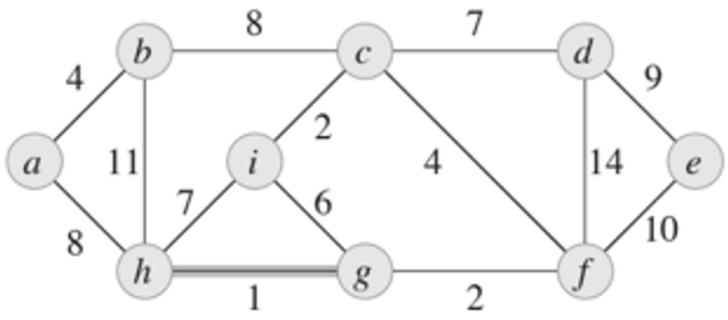
Brief Theory:

Kruskal's Algorithm is a greedy method to find the MST of a connected, weighted graph. It sorts all edges by weight and adds them to the MST in increasing order, ensuring no cycles.

```
MST-KRUSKAL(G, w)
1  A = ∅
2  for each vertex v ∈ G.V
3       MAKE-SET(v)
4  create a single list of the edges in G.E
5  sort the list of edges into monotonically increasing order by weight w
6  for each edge (u, v) taken from the sorted list in order
7       if FIND-SET(u) ≠ FIND-SET(v)
8            A = A ∪ {(u, v)}
9            UNION(u, v)
10 return A
```

Task:

1) Implement Kruskal's algorithm to find the MST(s) of the given connected, weighted graph using Adjacency matrix.



2) Detect cycles during edge selection.
3) Allow users to input edges and weights and prepare Linked Representation to further compute the MST using Kruskal's algorithm.
4) Add functionality to visualize the graph. (You may use Python)

Apparatus and components required: Computer with C or C++ Compiler and Linux/Windows platform.

Experimental/numerical procedure: Coding, compilation, editing, run and debugging.