Name: Ankush Singh
Group: 5C13
Roll No.: 040

Date: 10-10-2023

EXPERIMENT- 4

Aim:

To implement Minimum Spanning Tree and analyse its time complexity.

Code:

```
#include <bits/stdc++.h>
using namespace std;

struct Edge {
   int src, dest, weight;
};

bool compareEdges(const Edge& a, const Edge& b) {
   return a.weight < b.weight;
}

class Graph {
public:
   int V, E;
   vector<Edge> edges;
   Graph(int v, int e) {
      V = v;
      E = e;
   }
   void addEdge(int src, int dest, int weight) {
      Edge edge = {src, dest, weight};
      edges.push_back(edge);
   }
   int findParent(vector<int>& parent, int i) {
      if (parent[i] == -1)
         return i;
      return findParent(parent, parent[i]);
   }
   void kruskalMST() {
      vector<Edge> result;
      int i = 0;
      int edgeCount = 0;
      vector<int> parent(V, -1);
      int totalCost = 0;
      sort(edges.begin(), edges.end(), compareEdges);

      clock_t startTime = clock();
      while (edgeCount < V - 1 && i < E) {
         Edge nextEdge = edges[i];
         i++;
         int x = findParent(parent, nextEdge.src);
         int y = findParent(parent, nextEdge.dest);
```

```
            if (x != y) {
                result.push_back(nextEdge);
                edgeCount++;
                parent[x] = y;
                totalCost += nextEdge.weight; // Update the total cost
            }
        }
        clock_t endTime = clock();

        double executionTime = double(endTime - startTime) * 1000.0 / CLOCKS_PER_SEC;
        cout << "Edges in the Minimum Spanning Tree:" << endl;
        for (const Edge& edge : result) {
            cout << edge.src << " - " << edge.dest << " : " << edge.weight << endl;
        }
        cout << "Total Cost of Minimum Spanning Tree: " << totalCost << endl;
        cout << "Execution time: " << executionTime << " milliseconds" << endl;
    }
};

int main() {
    int V = 5;
    int E = 7;
    Graph graph(V, E);

    graph.addEdge(0, 1, 1);
    graph.addEdge(0, 2, 7);
    graph.addEdge(0, 3, 10);
    graph.addEdge(0, 4, 5);
    graph.addEdge(1, 2, 3);
    graph.addEdge(2, 3, 4);
    graph.addEdge(3, 4, 2);

    graph.kruskalMST();

    return 0;
}
```

Output:

```
Edges in the Minimum Spanning Tree:
0 - 1 : 1
3 - 4 : 2
1 - 2 : 3
2 - 3 : 4
Total Cost of Minimum Spanning Tree: 10
Execution time: 0.002 milliseconds
```