**Assignment #3**

1. Minimum Spanning Tree using Kruskal’s Algorithm

Student ID: 20210204039

Name: Abu Dojana

Group: A2

Date of Performance: 19. 12. 2023

Date of Submission: 03. 01. 2024

Date of Submission: 03. 01. 2024

Answer

#include <bits/stdc++.h>

using namespace std;

struct Edge {

    int u, v, w;

};

bool compareEdges(Edge a, Edge b) {

    return a.w < b.w;

}

vector<int> parent;

int findRepresentative(int u) {

    if (parent[u] == -1)

        return u;

    return findRepresentative(parent[u]);

}

void unionSets(int u, int v) {

    int repU = findRepresentative(u);

    int repV = findRepresentative(v);

    parent[repU] = repV;

}

vector<Edge> kruskalMST(int N, vector<Edge>& edges) {

    vector<Edge> minSpanningTree;

    sort(edges.begin(), edges.end(), compareEdges);

    parent.assign(N + 1, -1);

    for (int i = 0; i <= N; ++i) {

        parent[i] = -1;

    }

    for (Edge edge : edges) {

        int u = edge.u;

        int v = edge.v;

        if (findRepresentative(u) != findRepresentative(v)) {

            unionSets(u, v);

            minSpanningTree.push\_back(edge);

        }

    }

    return minSpanningTree;

}

int main() {

    int N, E;

    cout << "Enter the number of vertices: ";

    cin >> N;

    cout << "Enter the number of edges: ";

    cin >> E;

    vector<Edge> edges;

    cout << "Enter the edges and their weights(u, v, w):\n";

    cout << "+------+\n";

    cout << " Legend:" << endl;

    cout << "+------+\n";

    cout << "u = starting vertex\n";

    cout << "v = ending vertex\n";

    cout << "w = weight of the edge" << endl;

    cout << "Edge #: u v w" << endl << endl;

    for (int i = 0; i < E; ++i) {

        Edge edge;

        cout << "Edge " << i + 1 << ": ";

        cin >> edge.u >> edge.v >> edge.w;

        edges.push\_back(edge);

    }

    vector<Edge> minSpanningTree = kruskalMST(N, edges);

    cout << "\nThe edges of the minimum spanning tree are:\n";

    int totalWeight = 0;

    for (Edge edge : minSpanningTree) {

        cout << edge.u << " " << edge.v << " " << edge.w << endl;

        totalWeight += edge.w;

    }

    cout << "The total weight of the minimum spanning tree is: " << totalWeight << endl;

    return 0;

}

Console:

