Name- Manish Kumar

Enrollment No- 2020ITB007

Group – HX

Assignment 11: Kruskal's Algorithm

Implement Kruskal's  to find the MST of a randomly generated graph. Report and plot the # of comparisons against the size (# of edges) of the graph. Also plot the # of assignments against the # of edges.

#include <bits/stdc++.h>

using namespace std;

unordered\_map<int, int> vis;

int cnt = 1;

int comparison = 0;

int assignment = 5;

priority\_queue<pair<int, pair<int, int>>, vector<pair<int, pair<int, int>>>, greater<pair<int, pair<int, int>>>> q;

ofstream MyFile2("Text1.txt");

ofstream MyFile1("Text2.txt");

int ans = 0, e = 0;

int Kruskal(int Node)

{

    while (!q.empty())

    {

        comparison += 1;

        if (e < Node - 1)

        {

            comparison += 1;

            if (vis[q.top().second.first] == vis[q.top().second.second] &&

                vis[q.top().second.second] == 0)

            {

                ans += q.top().first;

                e++;

                vis[q.top().second.first] = cnt;

                vis[q.top().second.second] = cnt;

                cnt++;

                assignment += 2;

                comparison += 2;

            }

            else if (vis[q.top().second.first] != vis[q.top().second.second])

            {

                ans += q.top().first;

                assignment += 1;

                comparison += 1;

                if (vis[q.top().second.first] == 0)

                    vis[q.top().second.first] = vis[q.top().second.second];

                else if (vis[q.top().second.second] == 0)

                    vis[q.top().second.second] = vis[q.top().second.first];

                else if (vis[q.top().second.second] < vis[q.top().second.first])

                {

                    for (auto i : vis)

                    {

                        if (i.second == vis[q.top().second.first])

                            i.second = vis[q.top().second.second];

                    }

                }

                else

                {

                    for (auto i : vis)

                    {

                        if (i.second == vis[q.top().second.second])

                            i.second = vis[q.top().second.first];

                    }

                }

                assignment += 1;

                comparison += 1;

            }

            q.pop();

        }

        else

            break;

    }

    return ans;

}

int main()

{

int edges;

cin >> edges;

int node = 2 \* (edges -1);

for (int j = 0; j < edges; j++)

{

int u, v, w;

u = (rand() % node) + 1;

v = (rand() % node) + 1;

w = rand() % 10;

q.push(make\_pair(w, make\_pair(u, v)));

}

cout << Kruskal(node) << endl;

MyFile2 << edges << " " << comparison << endl;

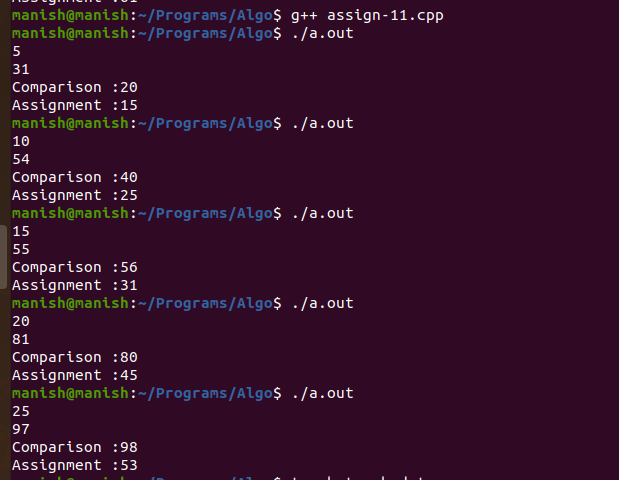
MyFile1 << edges << " " << assignment << endl;

cout << "Comparison :" << comparison << endl;

cout << "Assignment :" << assignment << endl;

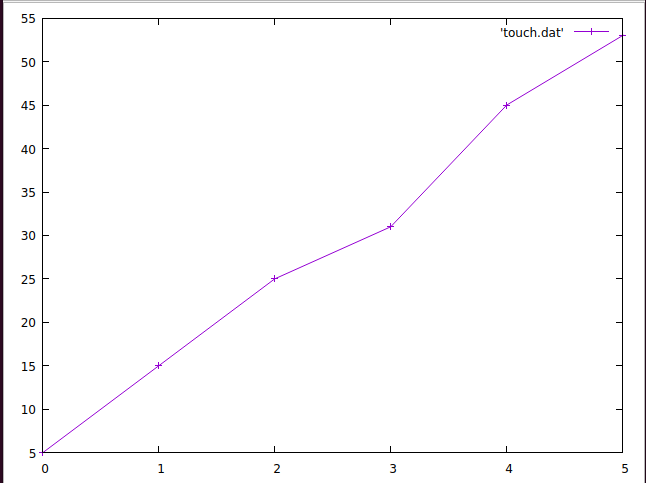
}

OUTPUT



PLOTTING GRAPH

a). Comparison



b). Assignment

