**Experiment 2.3**

**Student Name:   Gaurav Kumar                                UID: 22MCC20177**

**Branch:   CC-DevOps                                                     Section/Group:- 1/B**

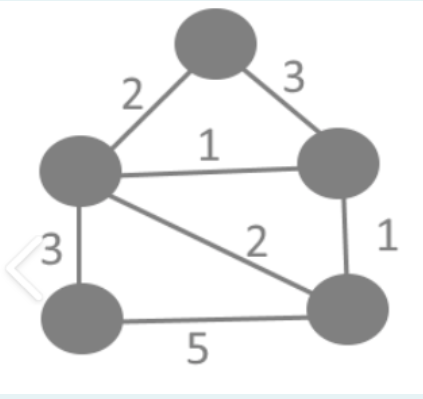
**Semester:   One                                                               Date of Performance: 29/11/2022**

**Subject Name:- Design & Analysis of Algorithms Lab                   Subject Code: 22CAP-646**

1. **Task to be done:**

Find Minimum Cost Spanning Tree of a given undirected graph

using Kruskal’s algorithm.



1. **Steps for experiment/practical: copy and paste your code here/screenshots**

#include <iostream>

#include <algorithm>

using namespace std;

const int MAX = 1e4 + 5;

int id[MAX], nodes, edges;

pair<long long, pair<int, int>> p[MAX];

void init()

{

for (int i = 0; i < MAX; ++i)

id[i] = i;

}

int root(int x)

{

while (id[x] != x)

{

id[x] = id[id[x]];

x = id[x];

}

return x;

}

void union1(int x, int y)

{

int p = root(x);

int q = root(y);

id[p] = id[q];

}

long long kruskal(pair<long long, pair<int, int>> p[])

{

int x, y;

long long cost, minimumCost = 0;

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

{

x = p[i].second.first;

y = p[i].second.second;

cost = p[i].first;

if (root(x) != root(y))

{

minimumCost += cost;

union1(x, y);

}

}

return minimumCost;

}

int main()

{

int x, y;

long long weight, cost, minimumCost;

init();

cout << "Enter Nodes and edges";

cin >> nodes >> edges;

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

{

cout << "Enter the value of X, Y and edges";

cin >> x >> y >> weight;

p[i] = make\_pair(weight, make\_pair(x, y));

}

sort(p, p + edges);

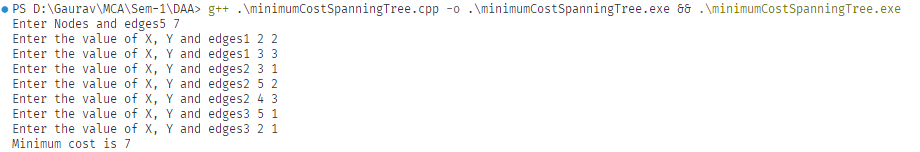
minimumCost = kruskal(p);

cout << "Minimum cost is " << minimumCost << endl;

return 0;

}

1. **Output (screenshots)**

****

**Evaluation Grid:**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Parameters | Marks Obtained | Maximum Marks |
| 1. | Demonstration and Performance  (Quiz) |  | 22 |
| 2. | Worksheet |  | 8 |