**Lab Assignment No-7(C)**

**PROBLEM STATEMENT:**

You have a business with several offices; you want to lease phone lines to connect them up

with each other; and the phone company charges different amounts of money to connect

different pairs of cities. You want a set of lines that connects all your offices with minimum total cost.

**Code --**

#include<iostream>

using namespace std;

class tree

{

int a[20][20], l, u, w, i, j, v, e, visited[20];

public:

void input();

void display();

void minimum();

};

void tree::input()

{

cout << "Enter the no. of branches: ";

cin >> v;

for (i = 0; i < v; i++)

{

visited[i] = 0;

for (j = 0; j < v; j++)

{

a[i][j] = 999; // Initially set all values to 999 (infinity)

}

}

cout << "\nEnter the no. of connections: ";

cin >> e;

for (i = 0; i < e; i++)

{

cout << "Enter the end branches of connections: " << endl;

cin >> l >> u;

cout << "Enter the phone company charges for this connection: ";

cin >> w;

a[l - 1][u - 1] = a[u - 1][l - 1] = w; // Set the weight for the connection

}

}

void tree::display()

{

cout << "\nAdjacency matrix:" << endl;

for (i = 0; i < v; i++)

{

for (j = 0; j < v; j++)

{

if (a[i][j] == 999) // If no connection, show 0 instead of 999

cout << "0 ";

else

cout << a[i][j] << " ";

}

cout << endl;

}

}

void tree::minimum()

{

int p = 0, q = 0, total = 0, min;

visited[0] = 1; // Start with node 0 (arbitrary starting point)

for (int count = 0; count < (v - 1); count++)

{

min = 999;

// Find the minimum edge among all the unvisited vertices

for (i = 0; i < v; i++)

{

if (visited[i] == 1)

{

for (j = 0; j < v; j++)

{

if (visited[j] != 1 && a[i][j] < min) // Avoid revisiting visited nodes

{

min = a[i][j];

p = i;

q = j;

}

}

}

}

// Mark the selected vertex as visited

visited[p] = 1;

visited[q] = 1;

// Add the minimum cost connection to the total

total += min;

// Output the minimum cost edge found in this iteration

cout << "Minimum cost connection is " << (p + 1) << " -> " << (q + 1) << " with charge: " << min << endl;

}

cout << "The minimum total cost of connections of all branches is: " << total << endl;

}

int main()

{

int ch;

tree t;

do

{

cout << "==========PRIM'S ALGORITHM=================" << endl;

cout << "\n1.INPUT\n \n2.DISPLAY\n \n3.MINIMUM\n" << endl;

cout << "Enter your choice: ";

cin >> ch;

switch (ch)

{

case 1:

cout << "\*\*\*\*\*\*\*INPUT YOUR VALUES\*\*\*\*\*\*\*" << endl;

t.input();

break;

case 2:

cout << "\*\*\*\*\*\*\*DISPLAY THE CONTENTS\*\*\*\*\*\*\*\*" << endl;

t.display();

break;

case 3:

cout << "\*\*\*\*\*\*\*\*\*MINIMUM\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

t.minimum();

break;

}

} while (ch != 4);

return 0;

}