Tarea 9 – Segunda Parte

Layla Tame A01192934

//

// main.cpp

// Tarea9.2

//

// Created by Layla Tame on 4/2/19.

// Copyright © 2019 Layla Tame. All rights reserved.

//

#include <iostream>

#include <fstream>

using namespace std;

int main()

{

//Read File

int iPuntos = 1, iDist;

string sArchivo;

ifstream fDistancias;

cout << "Nombre del archivo a leer: ";

cin >> sArchivo;

sArchivo = sArchivo + ".txt";

fDistancias.open(sArchivo);

fDistancias >> iPuntos;

int dist[iPuntos][iPuntos];

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

{

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

{

fDistancias >> iDist;

dist[i][j] = iDist;

}

}

fDistancias.close();

float costMetro;

cout << "Introduzca costo por metro: ";

cin >> costMetro;

//Prim

int caminos[iPuntos][iPuntos], nodos[iPuntos];

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

{

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

{

caminos[i][j] = 0;

}

nodos[i] = 0;

}

nodos[0] = 1;

int iSum = 0, node1 = -1, node2 = -1, iCount = iPuntos;

while(iCount-1 > 0)

{

int iMin = INT\_MAX;

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

{

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

{

if(dist[i][j]<iMin)

{

if((nodos[i] == 0 && nodos[j] == 1) || (nodos[j] == 0 && nodos[i] == 1))

{

iMin = dist[i][j];

node1 = i;

node2 = j;

}

}

}

}

nodos[node1] = nodos[node2] = 1;

caminos[node1][node2] = 1;

iSum += iMin;

iCount--;

}

cout << endl << "Costo minimo: " << iSum/costMetro << endl;

cout << "Conexiones:" << endl;

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

{

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

{

if(caminos[i][j] == 1)

{

cout << "(" << i << "," << j << ")" << endl;

}

}

}

int iX, iY;

cout << endl << "Introduzca dos puntos: ";

cin >> iX >> iY;

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

{

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

{

caminos[i][j] = 0;

}

nodos[i] = 0;

}

nodos[iX] = 1;

iSum = 0;

node1 = node2 = -1;

while(nodos[iY] != 1)

{

int iMin = INT\_MAX;

for(int i=iX; i<iPuntos; i++)

{

for(int j=iY; j<iPuntos; j++)

{

if(dist[i][j]<iMin)

{

if((nodos[i] == 0 && nodos[j] == 1) || (nodos[j] == 0 && nodos[i] == 1))

{

iMin = dist[i][j];

node1 = i;

node2 = j;

}

}

}

}

nodos[node1] = nodos[node2] = 1;

caminos[node1][node2] = 1;

iSum += iMin;

}

cout << "Distancia minima entre " << iX << " y " << iY << ": " << iSum << endl;

cout << "Conexiones:" << endl;

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

{

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

{

if(caminos[i][j] == 1)

{

cout << "(" << i << "," << j << ")" << endl;

}

}

}

}