Jacob Ciurca

Dr. Lu

Lab 11

Cosc 320

12/2/2020

**Pre-Lab:** Reviewed notes on spanning trees and short paths as well as performed minimum path algorithm on graphB.dat

**Lab:**

**Lab\_11.cpp:**

#include <iostream>

#include "d\_graph.h"

#include "d\_util.h"

#include <list>

#include <set>

#include <fstream>

using namespace std;

int main()

{

graph<string> g;

ifstream data;

data.open("graphB.dat");

int vertex;

int edge;

string a;

string e1;

string e2;

int e3;

data >> vertex;

for(int i=0; i<vertex; i++){

data >> a;

g.insertVertex(a);

}

data >> edge;

for(int i=0; i<edge; i++){

data >> e1;

data >> e2;

data >> e3;

g.insertEdge(e1, e2, e3);

}

cout << "Here is your starting graph" << endl;

cout << g;

cout << "Please enter a vertex from the above graph ";

string v;

cin >> v;

list<string> path;

list<string> testpath;

int pathsize;

pathsize = minimumPath(g, "A", v, path);

string pathvertex = "A";

if(pathsize < minimumPath(g, "B", v, testpath)){

pathsize = minimumPath(g, "B", v, path);

pathvertex = "B";

}

if(pathsize < minimumPath(g, "C", v, testpath)){

pathsize = minimumPath(g, "C", v, path);

pathvertex = "C";

}

if(pathsize < minimumPath(g, "D", v, testpath)){

pathsize = minimumPath(g, "D", v, path);

pathvertex = "D";

}

if(pathsize < minimumPath(g, "E", v, testpath)){

pathsize = minimumPath(g, "E", v, path);

pathvertex = "E";

}

if(pathsize < minimumPath(g, "F", v, testpath)){

pathsize = minimumPath(g, "F", v, path);

pathvertex = "F";

}

cout << "Vertex with largest minimum-path value: " << pathvertex << endl;

cout << "Minimum path value: " << pathsize << endl;

cout << "Minimum path is ";

list<string>::iterator iter;

for(iter = path.begin(); iter != path.end(); iter++){

cout << \*iter << " ";

}

return 0;

}

**graphB.dat:**

6

A B C D E F

10

A B 5

B A 5

B C 2

C B 2

C D 7

A D 15

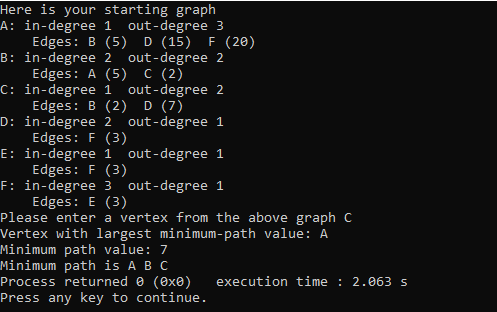
D F 3

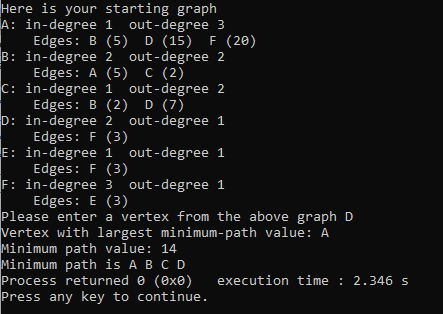
A F 20

E F 3

F E 3

**Sample Outputs:**





**Post-Lab:** This lab was good at giving us practice with minimum path tree traversal as well as teaching us how to use the algorithm and how it works. This lab took me about 1 hour 15 minutes to complete and I completed it by myself with no help.