**Lab 11**

**Name:** Will Townsend

**Class:** COSC 320 – Section 751

**Date:** 12/3/2020

**Lab Report:** Like all the other graph labs this was pretty straight forward as far as implementation. Conceptually I understand the minimum path algorithm, but there are still a few specifics I would like to look over in my personal studying time. This lab took me about an hour to complete. I can confirm that I did this 100% independently without help from outside resources (i.e. classmates or the internet).

**Prelab:**

A🡪B🡪C🡪D🡪F🡪E

Total Weight: 20

**Lab:**

**lab\_11.cpp:**

#include<stdio.h>

#include<fstream>

#include<utility>

#include<set>

#include<list>

#include"d\_graph.h"

#include"d\_util.h"

set<char> restOfVertices(char v,graph<char> g){

graph<char>::iterator curr = g.begin();

set<char> vertices;

for(int i=0;i<g.numberOfVertices();i++){

if(\*curr!=v)

vertices.insert(\*curr);

curr++;

}

return vertices;

}

int main(){

graph<char> graphB;

char input;

graphB.insertVertex('A');

graphB.insertVertex('B');

graphB.insertVertex('C');

graphB.insertVertex('D');

graphB.insertVertex('E');

graphB.insertVertex('F');

list<char> minList[graphB.numberOfVertices()];

ifstream fileB;

fileB.open("graphB.dat",std::fstream::in);

while (fileB){

char v1,v2;

int w;

fileB>>v1;

fileB>>v2;

fileB>>w;

graphB.insertEdge(v1,v2,w);

}

puts("Lab 11:");

printf("Pick a Vertex in Graph B: ");

std::cin>>input;

set<char> verts=restOfVertices(input,graphB);

int nums[verts.size()];

std::set<char>::iterator setNum = verts.begin();

int max=0;

char value;

for(int i=0;i<verts.size();i++){

nums[i]=minimumPath(graphB,input,\*setNum,minList[i]);

if(nums[i]>nums[max]){

max=i;

value=\*setNum;

}

setNum++;

}

std::cout<<"Largest minimum-path weight:V="<<value<<" & W=";

std::cout<<nums[max]<<std::endl;

std::cout<<"Path taken: ";

writeList(minList[max]," ");

return 0;

}

**Sample Outputs:**

Output 1:

Lab 11:

Pick a Vertex in Graph B: A

Largest minimum-path weight:V=E & W=20

Path taken: A B C D F E

Output 2:

Lab 11:

Pick a Vertex in Graph B: B

Largest minimum-path weight:V=E & W=15

Path taken: B C D F E

Output 3:

Lab 11:

Pick a Vertex in Graph B: C

Largest minimum-path weight:V=E & W=13

Path taken: C D F E

Output 4:

Lab 11:

Pick a Vertex in Graph B: D

Largest minimum-path weight:V=E & W=6

Path taken: D F E

Output 5:

Lab 11:

Pick a Vertex in Graph B: E

Largest minimum-path weight:V=F & W=3

Path taken: E F

Output 6:

Lab 11:

Pick a Vertex in Graph B: F

Largest minimum-path weight:V=E & W=3

Path taken: F E