Advanced Data Structures Assignment

Name : Shrirang R Mhalgi

Roll number : 222006

Class : SE B

Batch : B1

--------------------------------------------------------------------------------------------------------------------------------------

Problem Statement:

Tour operator organizes guided bus trips across the Maharashtra. Tourists may have different preferences. Tour operator offers a choice from many different routes. Every day the bus moves from starting city S to another city F as chosen by client. On this way, the tourists can see the sights alongside the route travelled from S to F. Client may have preference to choose route. There is a restriction on the routes that the tourists may choose from, the bus has to take a short route from S to F or a route having one distance unit longer than the minimal distance. Two routes from S to F are considered different if there is at least one road from a city A to a city B which is part of one route, but not of the other route.

Code:

#include<iostream>

#include<stack>

using namespace std;

int inf=9999;

int i,j,n,e;

char s;

class graph{

int vis[10];

int dist[10];

char par[10];

int mat[10][10];

public:

void init(){

for(i=0;i<10;i++)

{

vis[i]=0;

dist[i]=inf;

par[i]='-';

}

}

void accept();

void display();

void dj();

int visited();

};

void graph :: accept(){

char src,dest,wt;

cout<<"Enter the number of nodes in graph : ";

cin>>n;

cout<<"Enter the number of edges in graph : ";

cin>>e;

cout<<"Enter the matrix row-wise\n";

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

{

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

{

cin>>mat[i][j];

}

}

}

void graph :: dj(){

init();

cout<<"Enter the source vertex : ";

cin>>s;

int src=s-65;

s=src;

dist[src]=0;

while(!visited())

{

vis[src]=1;

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

{

int newd = (mat[src][i]==0 && src!=i) ? inf : min(dist[src]+mat[src][i],dist[i]);

//if(newd<dist[i])

{

dist[i]=newd;

par[i]=src;

}

}

int mini=999;

for(int x=0;x<n;x++)

{

if(vis[x]==0 && mini>dist[x])

{

mini=dist[x];

src=x;

}

}

}

}

void graph :: display(){

char d;

cout<<"Enter the destination : ";

//ws(cin);

// cout<<"i am";

cin>>d;

int des=d-65;

if(des>=n)

cout<<"Invalid destination\n";

if(des==s)

cout<<"Source and destination are same\n";

else{

cout<<"\nFinal value of path is : "<<dist[des];

cout<<"\nPath is : "<<(char)(des+65);

stack <int> s1;

for(int i=des;i!=s;)

{

s1.push(i);

i=par[i];

}

while(!s1.empty())

{

cout<<(char)(s1.top()+65);

s1.pop();

}

}

}

int graph :: visited(){

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

{

if(vis[i]==0)

return 0;

}

return 1;

}

int main(){

char ch;

graph o;

o.accept();

do{

o.dj();

o.display();

}while(ch=='y'||ch=='Y');

return 0;

}

Output:

Enter the number of nodes in graph : 4

Enter the number of edges in graph : 5

Enter the matrix row-wise

0 10 5 15

10 0 20 0

5 20 0 16

15 0 16 0

Enter the source vertex : B

Enter the destination vertex : C

Minimum distance : 15

Path : BAC