**中国海洋大学计算机科学与技术系**

**实验报告**

**姓名：徐云哲 学号：15020031088 专业：计算机科学与技术**

**实验题目：用迪杰斯特拉算法编写一个校园导游图 教师：魏振钢**

**一、实验目的：**

用迪杰斯特拉算法思想实现校园导游图，为游客提供一条从当前位置到目的地的最短路径

1. **实验思路**

Step1: 用邻接矩阵作为校园平面图的存储结构，每两个景点之间的距离作为边的权值，首先定义一个初始化函数，初始化邻接矩阵

Step2: 定义三个一维数组，一个bool类型数组final[]用来记录从v0到vi是否已经确定了最短路径，是则记final[i]=true,否记final[i]= flase；一个int 类型数组P[]用来记录从v0到vi的当前最短路径上的vi的直接前驱顶点编号，若v 到vi之间有边则记P[i] = v的编号，否则记P[i] = -1；最后一个数组D[]用来记录从v0到vi之间的最短路径长度，存在则记v0到vi之间边的权值或者权值和，否则记为INFINITY

Step3:定义一个求最短路径的函数，传入的参数为图、起点、数组P、数组D

然后进行n-1次for循环，找出vo到其余n-1个顶点之间的最短路径，比较当前D[]数组中最小值，找到最小值的编号v，该编号就是从v0出发到所有顶点中距离最短的顶点编号，然后把final[v]的值置为true。说明从v0出发到顶点v已经找到最短路径；

Step4:更新D[]数组

Step5:循环2、5步得到数组D

1. **实验内容**

#include<iostream>

#include<stdlib.h>

#include<string>

#define INFINITY 100000

#define MAX\_VERTEX\_NUM 20

using namespace std;

typedef struct ArcCell

{

int adj;

} ArcCell,AdjMatrix[MAX\_VERTEX\_NUM][MAX\_VERTEX\_NUM];

typedef struct

{

int vexs[MAX\_VERTEX\_NUM];

AdjMatrix arcs;

int vexnum,arcnum;

} MGraph;

MGraph G;//把图定义为全局变量

**void CreateUDN(MGraph &G)**

{

//采用邻接矩阵法构造无向网G

G.vexnum=15;

int i,j;

for(i=0; i<G.vexnum; ++i)

G.vexs[i]=i;//将1-15个顶点向量存入数组

for(i=0; i<G.vexnum; ++i)

for(j=0; j<G.vexnum; ++j)

G.arcs[i][j].adj=INFINITY;//初始化邻接矩阵，INFINITY表示两点之间无法到达

//将校园路线的邻接矩阵输入

for(i=0; i<G.vexnum; ++i)

G.arcs[i][i].adj=0;

G.arcs[1][0].adj=10;

G.arcs[2][0].adj=310;

G.arcs[2][1].adj=310;

G.arcs[3][2].adj=10;

G.arcs[4][2].adj=50;

G.arcs[5][2].adj=500;

G.arcs[5][4].adj=600;

G.arcs[6][5].adj=200;

G.arcs[7][3].adj=100;

G.arcs[8][2].adj=70;

G.arcs[8][4].adj=100;

G.arcs[9][2].adj=500;

G.arcs[9][8].adj=400;

G.arcs[10][3].adj=250;

G.arcs[10][7].adj=200;

G.arcs[11][2].adj=800;

G.arcs[11][8].adj=600;

G.arcs[12][10].adj=400;

G.arcs[13][2].adj=1000;

G.arcs[13][12].adj=350;

G.arcs[14][2].adj=1500;

G.arcs[14][11].adj=500;

G.arcs[14][12].adj=300;

G.arcs[14][13].adj=150;

for(i=0; i<G.vexnum; ++i)

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

G.arcs[j][i].adj= G.arcs[i][j].adj;

}

**void ShortestPath\_DIJ(MGraph G,int v0,int p[][MAX\_VERTEX\_NUM],int D[])**

{

//S集为已求得的最短路径的终点集合，设置一个数组final,当final[v]=true时，即已经求得从v0到v的最短路径

//D数组记录弧的权值

int i,j,k,w,v,min;

bool final[15];

for(v=0; v<G.vexnum; v++) //初始化

{

final[v]=false;

D[v]=G.arcs[v0][v].adj;

for(w=0; w<G.vexnum; w++)

p[v][w]=-1;

if(D[v]<INFINITY)

{

p[v][0]=v0;

p[v][1]=v;

}

}

D[v0]=0;

final[v0]=true;

//开始主循环

for(i=0; i<G.vexnum; ++i)

{

min=INFINITY;

for(w=0; w<G.vexnum; w++)

{

if(!final[w])

if(D[w]<min)

{

v=w;

min=D[w];

}

}

final[v]=true;//离v0最近的v加入S集

for(w=0; w<G.vexnum; w++) //更新当前路径

{

if(!final[w] && (min+G.arcs[v][w].adj < D[w]))

{

D[w]=min+G.arcs[v][w].adj;

for(j=0; j<G.vexnum; j++)

{

p[w][j]=p[v][j];

if(p[w][j]==-1)//在p[w][]第一个等于-1的地方加上顶点w

{

p[w][j]=w;

break;

}

}

}

}

}//for

}

int main()

{

int ch1,ch2,j,n;

int D[15];

int p[MAX\_VERTEX\_NUM][MAX\_VERTEX\_NUM];

CreateUDN(G);

while(1)

{

cout <<endl;

cout <<"\*\*\*\*\*\*\*\*\*\* 欢迎来到中国海洋大学校园导游系统！\*\*\*\*\*\*\*\*\*\*"<<endl;

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

cout <<"\*\*\*\*\*\*\*\*\*\* 1.景点信息查询\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;//1、2系统还没有完善

cout <<"\*\*\*\*\*\*\*\*\*\* 2.显示景点视图\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout <<"\*\*\*\*\*\*\*\*\*\* 3.最短路径查询\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout <<"\*\*\*\*\*\*\*\*\*\* 0.退出系统\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

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

cout <<"请选择："<<endl;

cin >> n;

if(n==3)

{

cout <<"\*\*\*\*\*\*\*\*\*\* 以下是我们的校园景点编号 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" <<endl;

cout <<"0.北门 1.校医院 2.北区宿舍楼 3.北区操场 4.北区食堂 5.东区操场"<<endl;

cout <<"6.东门 7.大学生活动中心 8.第三教学区 9.屋子顶 10.西门"<<endl;

cout <<"11.信息学院 12.图书馆 13.行远楼 14.南门 "<<endl;

cout <<"\*\*\*\*\*\*\*\*\*\* 请输入你的当前位置 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cin >> ch1;

cout <<"\*\*\*\*\*\*\*\*\*\* 请输入你的目的地 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cin >> ch2;

ShortestPath\_DIJ(G,ch1,p,D);

if(D[ch2]==-1)

{

cout<<ch1<<"到"<<ch2<<"之间无最短路径！"<<endl;

exit(0);

}

cout<< ch1 <<"到"<< ch2 <<"的最短距离为："<< D[ch2]<<endl;

cout <<"最短路径为："<<endl;

for(j=0; j<G.vexnum; j++)

{

if(p[ch2][j]>-1)

cout<<G.vexs[p[ch2][j]]<<"-->";

}

cout <<endl;

}

else if(n==0)

{

cout << ">>>退出成功！" << endl;

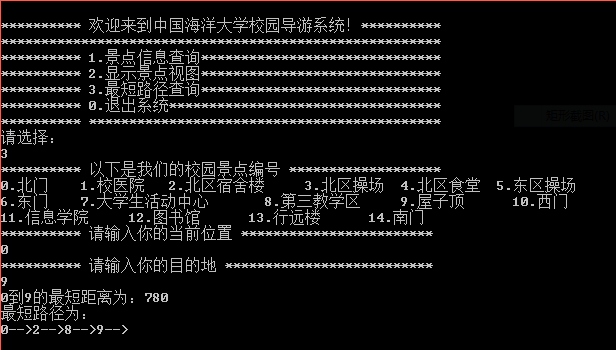
exit(0);

}

}

}

1. **程序运行结果**

****