1. **需求分析**

（1）来访客人到学校参观的时候需要有一个学校导游系统，可以通过这个导航系统了解学校内的各个景点位置、介绍以及去景点的最佳路径。

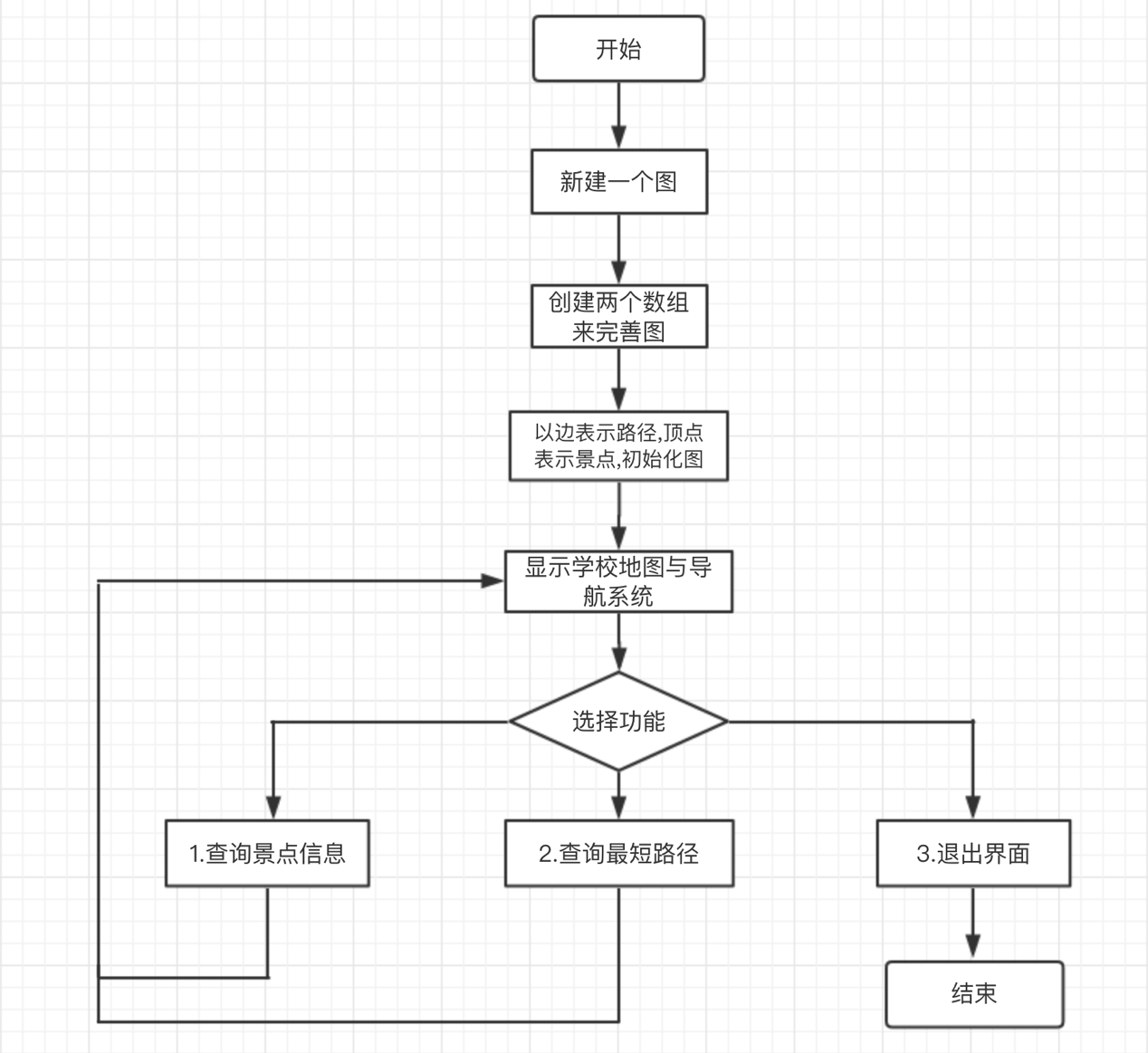
1. 功能

在系统首页显示一个导航图，校园景区地图，这个导航中给用户提供查询景点、浏览路径、寻找最佳的方案到达目的地以及最佳路径的路径及长度。

1. 实现的目标

实现对河海大学江宁校区的校园导游系统。

1. **概要设计**
2. 流程图

（2）概要设计

1. 定义三个结构体，第一个结构体储存顶点的具体信息，第二个结构体储存顶点的编号以及景点之间的距离，第三个储存两个景点之间的最短路径以及最短路径距离。

typedef struct{

char info[200]; //景点介绍

}vexnode;

typedef struct{

vexnode vexs[10]; //景点编号

int arcs[vn][vn]; //各个景点之间的距离

}GraphMatricx;

typedef GraphMatricx \*pgraphmatricx;

typedef struct{

int len; //两个景点之间最短路径长度

int prevex; //从Vi到Vj的最短路径上Vj的前一个景点

}path;

path dist[vn]; //Vi到其他所有景点的最短路径长度

1. 定义三个数组用来存放信息，vexs[10]数组用来存放景点编号，arcs[10][10]数组用来存放各个景点的距离，dist[vn]数组用来存放两个景点之间的最短路径长度
2. 在主函数中调用map()函数显示导航系统界面以及学校地图
3. initInfo()函数完成景点之间的距离以及景点信息的初始化
4. searchInfo()函数用来查询各个景点的具体介绍
5. display()函数用来展示系统界面
6. 核心算法dijkstra()函数

void dijkstra(GraphMatricx \*pgraph,path dist[],int start){

int i,j,mv;

int minw;

for(i = 1; i < vn; i++){

minw = MAX;

mv = start;

for(j = 0; j < vn; j++){

if(pgraph->arcs[j][j] == 0 && dist[j].len < minw){

mv = j;

minw = dist[j].len;

}

}

if(mv == start){

break;

}

pgraph->arcs[mv][mv] = 1;

for(j = 0; j < vn; j++){

if(pgraph->arcs[j][j] == 0 && dist[j].len > dist[mv].len + pgraph->arcs[mv][j]){

dist[j].prevex = mv;

dist[j].len = dist[mv].len + pgraph->arcs[mv][j];

}

}

}

}

算法分析：算法中求最短路径由一个大循环组成，其中外循环运行n次，内循环为两个，也均运行n次，因此算法的时间复杂度为O(n2)。

**3、详细设计**

/\*

2、校园导游咨询（为来访的客人提供各种信息服务）

1）参考校园设计平面图，在校园景点选10个左右景点。以图中顶点表示校园内各景点，存放景点名称、代号、简介等信息；以边表示路径，存放路径长度等有关信息。

2）为来访客人提供图中任意景点相关信息的查询。

3）为来访客人提供任意景点的问路查询，即查询任意两个景点之间的一条最短路径。

\*/

#include<iostream>

using namespace std;

#define vn 10

#define MAX 99999

typedef struct{

char info[200]; //景点介绍

}vexnode;

typedef struct{

vexnode vexs[10]; //景点编号

int arcs[vn][vn]; //各个景点之间的距离

}GraphMatricx;

typedef GraphMatricx \*pgraphmatricx;

typedef struct{

int len; //两个景点之间最短路径长度

int prevex; //从Vi到Vj的最短路径上Vj的前一个景点

}path;

path dist[vn]; //Vi到其他所有景点的最短路径长度

void display(GraphMatricx \*pgraph,int arcs[][vn]);

void init(GraphMatricx \* pgraph,path dist[],int start); //完成dist数组的初始化

void dijkstra(GraphMatricx \*pgraph,path dist[],int start); //计算一个景点到其他景点的最短路径

void map(); //显示景点地图

void searchInfo(GraphMatricx \*pgraph,int a); //查询景点信息

void initInfo(GraphMatricx \*pgraph,int arcs[vn][vn]); //景点信息初始化

int main(){

pgraphmatricx pgraph = new GraphMatricx;

//景点之间的距离

int arcs[vn][vn] = {

{ 0, 10, 30, 40, 40, MAX, MAX, MAX, MAX, MAX },

{ 10, 0, 15, 10, MAX, MAX, MAX, MAX, MAX, MAX },

{ 30, 15, 0, 15, 10, MAX, MAX, MAX, MAX, MAX },

{ 40, 10, 15, 0, 15, MAX, MAX, MAX, MAX, MAX },

{ 40, MAX, 10, 15, 0, 15, MAX, MAX, MAX, MAX },

{ MAX, MAX, MAX, MAX, 15, 0, 30, MAX, MAX,MAX},

{ MAX, MAX, MAX, MAX, MAX, 30, 0, MAX, 15, MAX },

{ MAX, MAX, MAX, MAX, MAX, MAX, MAX, 0, MAX, 30 },

{ MAX, MAX, MAX, MAX, MAX, MAX, 15, MAX, 0, 20 },

{ MAX, MAX, MAX, MAX, MAX, MAX, MAX, 30, 20, 0 }

};

display(pgraph,arcs);

return 0;

}

void display(GraphMatricx \*pgraph,int arcs[vn][vn]){

int choice; //选择项

int start,end; //查询的起点与终点

int route[vn]; //查询的终点及它的所有前驱节点

do{

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

route[i] = -1;

}

initInfo(pgraph,arcs);

map();

cout<<"1.景点信息查询"<<endl;

cout<<"2.问路查询"<<endl;

cout<<"3.退出系统"<<endl;

cin>>choice;

if(choice == 1){

searchInfo(pgraph,choice);

}

if(choice == 2){

cout<<"请分别输入起点和终点的序号"<<endl;

cin>>start >>end;

init(pgraph,dist,start-1);

dijkstra(pgraph,dist,start-1);

if(dist[end-1].prevex == -1){

cout<<"两个景点没有通路"<<endl;

continue;

}

cout<<"这两个景点的最短路径为:"<<endl;

route[0] = end;

for(int i = 1; i < vn; i++){

if(dist[end-1].prevex != start-1){

route[i] = dist[end-1].prevex+1;

dist[end-1].prevex = dist[dist[end-1].prevex].prevex;

}

else{

route[i] = dist[end-1].prevex+1;

break;

}

}

for(int i = vn-1; i >0; i--){

if(route[i] != -1){

cout<<route[i];

cout<<"->";

}

else{

continue;

}

}

cout<<route[0]<<endl;

cout<<"这两个景点的最短路径距离为:"<<endl;

cout<<dist[end-1].len<<"米"<<endl;

}

}while(choice != 3);

}

void init(GraphMatricx \* pgraph,path dist[],int start){

int i;

pgraph->arcs[start][start] = 1;

for(i = 0; i < vn; i++){

dist[i].len = pgraph->arcs[start][i];

if(dist[i].len != MAX){

dist[i].prevex = start;

}

else{

dist[i].prevex = -1;

}

}

}

void dijkstra(GraphMatricx \*pgraph,path dist[],int start){

int i,j,mv;

int minw;

for(i = 1; i < vn; i++){

minw = MAX;

mv = start;

for(j = 0; j < vn; j++){

if(pgraph->arcs[j][j] == 0 && dist[j].len < minw){

mv = j;

minw = dist[j].len;

}

}

if(mv == start){

break;

}

pgraph->arcs[mv][mv] = 1;

for(j = 0; j < vn; j++){

if(pgraph->arcs[j][j] == 0 && dist[j].len > dist[mv].len + pgraph->arcs[mv][j]){

dist[j].prevex = mv;

dist[j].len = dist[mv].len + pgraph->arcs[mv][j];

}

}

}

}

void map(){

cout << endl;

cout << " ★---------★---------★---------★---------★---------★---------★" << endl;

cout << " 【10】叠翠山" << endl;

cout << " ☆ ☆" << endl;

cout << " 【9】宿舍区 【8】文体中心 " << endl;

cout << " " << endl;

cout << " ☆ 【7】西操场 【6】小浪底 ☆" << endl;

cout << " " << endl;

cout << " 【5】图书馆 " << endl;

cout << " ☆ 【4】勤学楼 ☆" << endl;

cout << " 北 【3】三峡广场 " << endl;

cout << " 西<-|->东 " << endl;

cout << " ☆ 南 【2】博学楼 ☆" << endl;

cout << " 【1】南大门 " << endl;

cout << " ★---------★---------★---------★---------★---------★---------★" << endl;

}

void searchInfo(GraphMatricx \*pgraph,int a){

cout<<"请输入要查询的景点信息"<<endl;

scanf("%d",&a);

switch(a){

case 1:cout<<pgraph->vexs[0].info<<endl;

cout<<endl;

break;

case 2:cout<<pgraph->vexs[1].info<<endl;

cout<<endl;

break;

case 3:cout<<pgraph->vexs[2].info<<endl;

cout<<endl;

break;

case 4:cout<<pgraph->vexs[3].info<<endl;

cout<<endl;

break;

case 5:cout<<pgraph->vexs[4].info<<endl;

cout<<endl;

break;

case 6:cout<<pgraph->vexs[5].info<<endl;

cout<<endl;

break;

case 7:cout<<pgraph->vexs[6].info<<endl;

cout<<endl;

break;

case 8:cout<<pgraph->vexs[7].info<<endl;

cout<<endl;

break;

case 9:cout<<pgraph->vexs[8].info<<endl;

cout<<endl;

break;

case 10:cout<<pgraph->vexs[9].info<<endl;

cout<<endl;

break;

default:cout<<"请输入有效值!"<<endl;

cout<<endl;

break;

}

}

void initInfo(GraphMatricx \*pgraph,int arcs[vn][vn]){

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

for(int j = 0; j < 10; j++){

pgraph->arcs[i][j] = arcs[i][j];

}

strcpy(pgraph->vexs[0].info,"【南大门】南大门是河海大学正门，上书邓小平同志亲笔所题校名。");

strcpy(pgraph->vexs[1].info,"【博学楼】博学楼是江宁校区最高建筑，是商学院的院部和教学楼，环境优雅，设施完善。");

strcpy(pgraph->vexs[2].info,"【三峡广场】位于图书馆于南门之间的草坪，气势雄伟，设有地下停车场，2005年温总理视察河海大学在此发表讲话。" );

strcpy(pgraph->vexs[3].info,"【勤学楼】勤学楼由五个部分组成，是能电院计信院大禹院的院部以及科研办公楼，也是学校基建处办公地。");

strcpy(pgraph->vexs[4].info,"【图书馆】图书馆收纳了各个专业的相关书籍，是名副其实的“书的海洋”，提供了丰富的学习资源。");

strcpy(pgraph->vexs[5].info,"【小浪底】小浪底音乐广场由半圆形看台回音壁组成，常常举办音乐会嘉年华等室外文娱活动，也是江宁校区地标。");

strcpy(pgraph->vexs[6].info,"【西操场】西操场为临近宿舍区的塑胶足球场，周长300米，常有同学在此晨练夜跑，周末常常举办足球赛和各类素拓。");

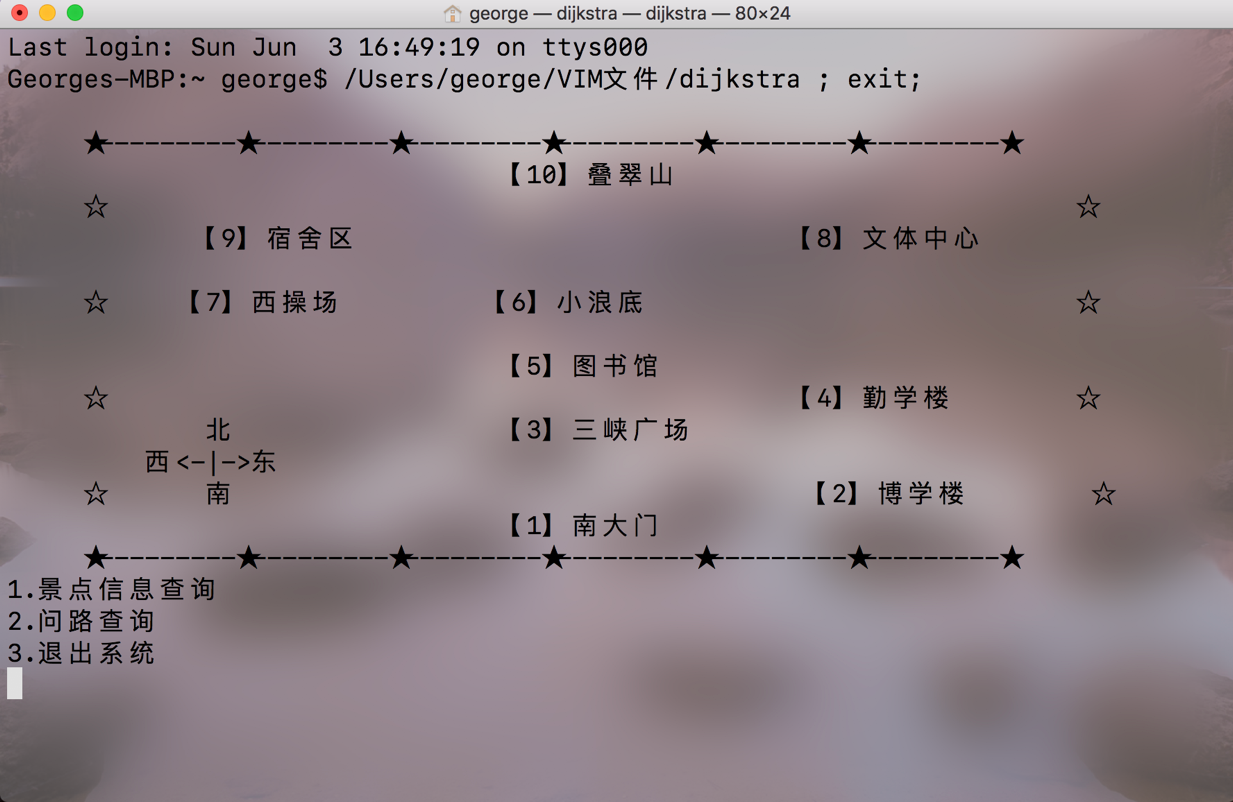
strcpy(pgraph->vexs[7].info,"【文体中心】文体中心共有4层，内设有乒乓球，篮球，羽毛球等场地，同时也是重大典礼举办的场所。");

strcpy(pgraph->vexs[8].info,"【宿舍区】宿舍区由新旧两部分学生组团组成，建有研究生与本科生宿舍楼及其配套的一体化餐饮中心。");

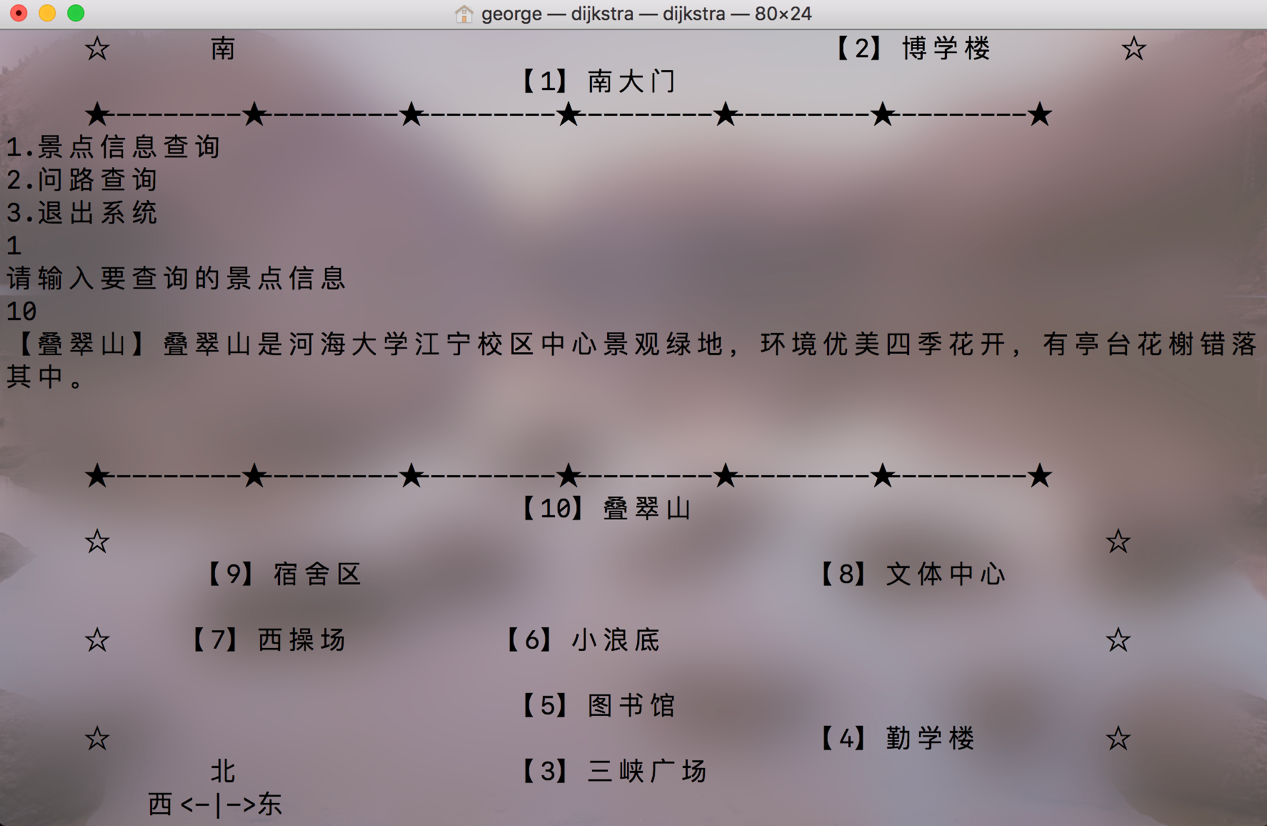
strcpy(pgraph->vexs[9].info,"【叠翠山】叠翠山是河海大学江宁校区中心景观绿地，环境优美四季花开，有亭台花榭错落其中。");}

1. **调试测试**

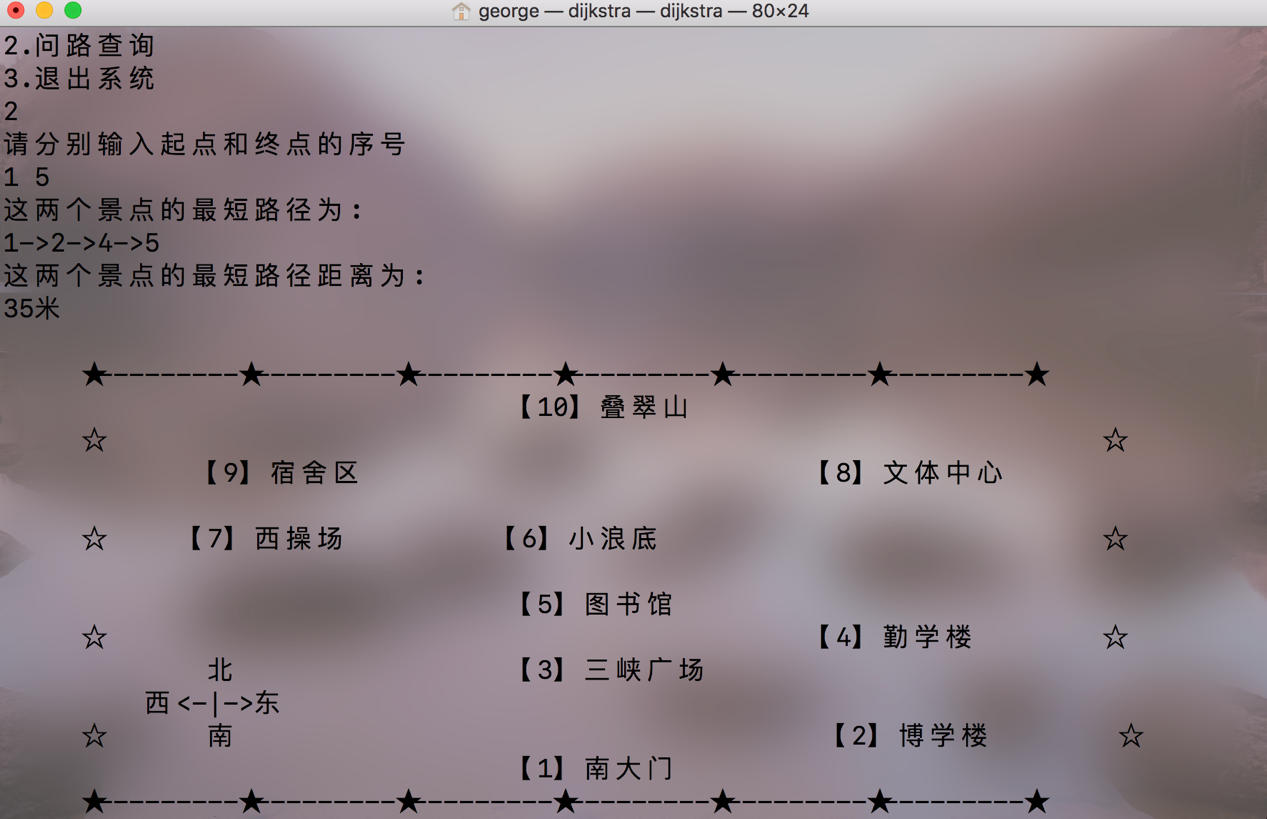
（1）这是主界面



1. 接下来介绍第一个功能，输入“1”之后，再输入对应想要查询的景点编号，即可获得景点信息。



1. 第二个功能输入“2”即可 再输入想要去的两个景点，即可知道路径信息以及最佳路径



1. 功能可以在同一个页面里使用，输入“3”可以退出系统，继续输入“1”或者“2”可以继续查询。

