
琼州学院电子信息工程学院

课程设计报告

课程名称：《数据结构》课程设计

设计题目：校园导游咨询

专 业： 软件工程

班 级： 2010 软件工程

学生姓名：

学 号：

起止日期：

指导教师：

注
事

一、
计
的

意
项

设
目

指导教师评语：

最终成绩：

指导教师签名：

年 月 日

成 绩 评 定	项 目	权 重	成 绩
	1、设计过程中的学习态度	0.2	
	2、课程设计的质量及答辩	0.5	
	3、设计报告书规范程度 <small>-可编辑修改-</small>	0.3	
	4、总成绩		

《数据结构》是一门实践性较强的软件基础课，为了学好这门课程，必须在掌握理论知识的同时，加强上机实践。本课程设计的目的就是要达到理论与实际应用相结合，使同学们能够根据数据对象的特性，学会数据组织的方法，能把现实世界中的实际问题在计算机内部表示出来，并培养基本的、良好的程序设计技能。

二、设计要求

1. 通过这次课程设计，要求在数据结构的逻辑特性和物理表示、数据结构的选择应用、算法的设计及其实现等方面加深课程基本内容的理解。同时，在程序设计方法以及上机操作等基本技能和科学作风方面受到比较系统和严格的训练。

2. 学生必须仔细阅读《数据结构》课程设计要求，以学生自学为主、指导教师指导为辅，独立完成课程设计的任务，有问题及时主动与指导教师沟通。

3. 本次课程设计按照教学要求需要在本学期 15 周前完成，学生要发挥自主学习的能力，充分利用时间，安排好课程设计的时间计划，并在课程设计过程中不断检测自己的计划完成情况，及时向指导教师汇报。

4. 编程语言：C 语言。

三、课程设计说明书的格式要求

设计文档的撰写必须提前进行，以保证使文档与程序同步提交。

- | | |
|------------|-----------------|
| 1. 设计题目 | 2. 运行环境（软、硬件环境） |
| 3. 算法的需求分析 | 4. 算法概要设计 |
| 5. 算法详细设计 | 6. 算法的测试 |
| 7. 运行结果分析 | 8. 收获及体会 |

四、问题分析、设计和测试过程要规范化

1. 需求分析：将题目中要求的功能进行叙述分析。

2. 概要设计：算法的设计说明，描述解决此问题的数据存储结构，（有些题目已经指定了数据存储的，按照指定的设计），描述算法建议使用流程图，进行算法分析指明关键语句的时间复杂度。

3. 详细设计：即各个算法的具体实现步骤，每个题目要有相应的源程序，其中每个功能模块采用不同的函数实现。源程序要规范编写：结构要清晰，注释要清楚。重点函数的重点变量和重点功能部分要加上清楚的程序注释。

4. 调试和测试：给出实现功能的一组或多组测试数据，程序调试后，将按照此测试数据进行测试的结果列出来。在调试过程中遇到的问题 and 解决方法也要记录下来。程序要能够正常运行，还要有基本的容错功能。尽量避免出现操作错误时出现死循环。

5. 改进措施：对有些题目提出算法改进方案，比较不同算法的优缺点。

五、对指导教师的要求

指导教师要关心学生的课程设计进展，认真答疑。对课程设计报告的撰写要给予充分的指导，报告中切忌出现大篇源代码，应严格要求学生将主要篇幅放在“原理实现”上，即如何用框图表达设计和实施思想。课程设计报告要用红笔批阅，最终成绩以优、良、中、及格与不及格分等计算。

目录

摘要.....	1.....
1 设计内容和要求.....	
1.1 设计内容	
1.1 设计要求	
2 概要设计.....	2.....
2.1 程序的模块图	2.....
2.2 主函数的概要设计.....	3.....
2.3 查找介绍函数的概要设计.....	3.....
2.4 查找最短路径函数的概要设计.....	3.....
2.5 景点分布图的概要设计.....	4.....
2.6 退出函数的概要设计	4.....
3 详细设计.....	4.....
3.1 程序的流程图	4.....
3.2 主函数的详细设计.....	5.....
3.3 查找介绍函数的详细设计.....	6.....
3.4 查找最短路径函数的详细设计.....	6.....
3.5 景点分布图的详细设计.....	9.....
3.6 退出函数的详细设计	10.....
3.7 数据结构的详细设计	10.....
4 软件测试.....	11.....
4.1 菜单的测试.....	11.....

4.2 查找景点简介的测试	12
4.3 查找两个景点之间的最短距离的测试.....	14
4.4 查看景点分布图的测试.....	15
4.5 退出的测试.....	16
5 软件使用说明.....	18
6 参考文献.....	19
7 附录	20
7.1 系统完整代码	20

摘要

现代快节奏的生活使得都市人越来越渴望亲近自然，因此外出旅游现在被越来越多的都市人所看中，所以如何快速方便的找到我们想要的旅游景点的信息和最短路径就成了一个很重要的问题。

本设计基于图的结构，创建一个无向图，针对游客的实际需求，将琼州学院的景点编号、名称、介绍等信息放入到图的顶点当中并保存在景点文本文件当中，将两个景点的编号和它们之间的距离当作权值也保存到权值文本文件当中，利用迪杰斯特拉算法来求从一个景点到另一个景点的最短距离，利用 `Search()` 函数来查找景点，并显示出它的信息，从而解决了要查找景点信息和景点之间的最短路径的问题，最后按照显示屏上的提示进行相关的操作。

关键词：分布图、查找信息、最短距离、校园导游咨询

1 设计内容和要求

1.1 设计内容

依据课程设计的要求，利用一个无向图的结构，将景点当作图的顶点，将景点之间的距离当作权值来储存，然后根据游客自己的需求，按照显示屏上的提示来进行查找景点介绍，查找两个景点之间的最短距离，退出程序等基本操作。

1.1 设计要求

本软件为校园导游咨询系统，根据游客的实际需求而设计，首先创建一个无向图，然后从文件当中读取所有景点的编号、名称、介绍和两点之间的权值，并将它们写入到无向图当中。功能主要包括查找已知景点的信息，查找从一个景点到另一个景点的最短路径，退出等基本操作。

软件的界面要求使用 VC++6.0 的运行环境。

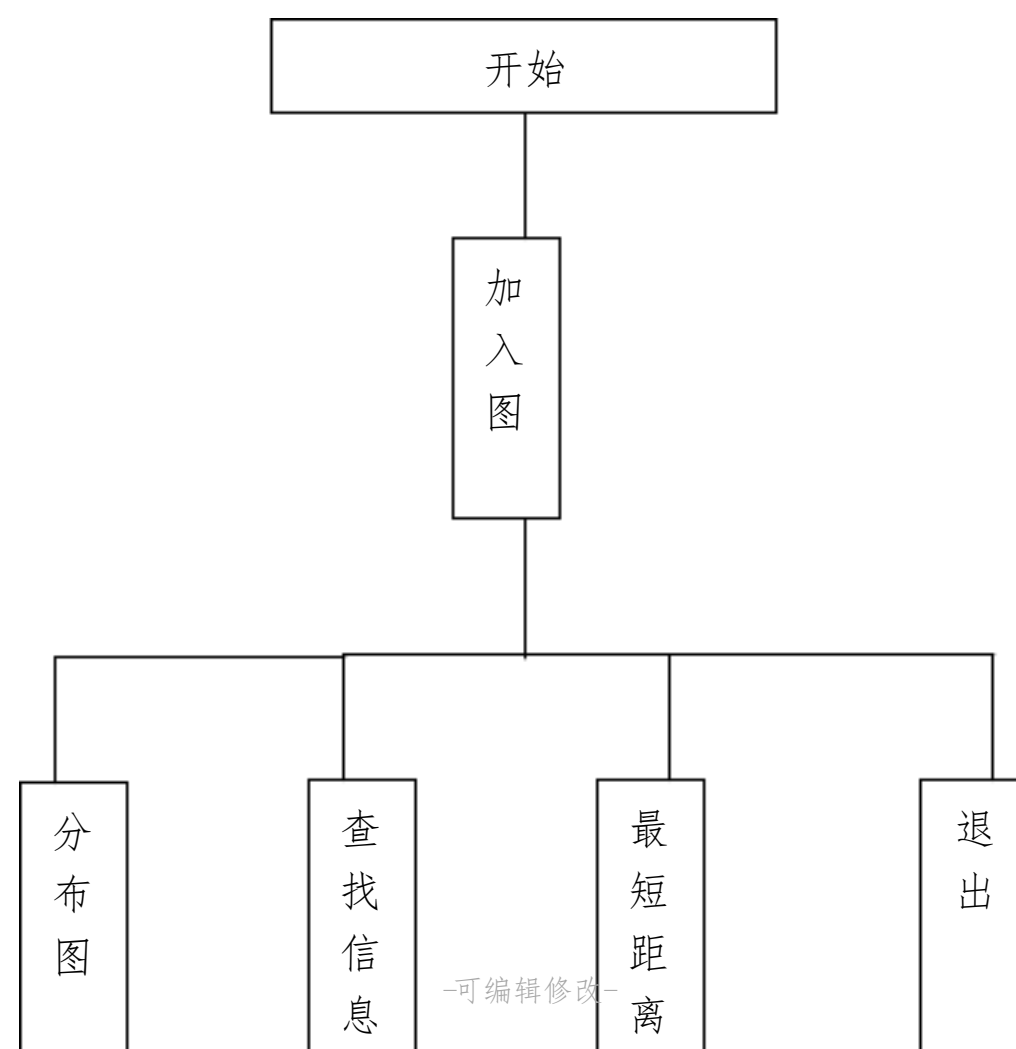
软件的数据库包括校园景点的编号、名称、介绍和两个景点之间的距离（权值），首先要定义顶点的数据类型结构体，里面包括景点的编号、名称、介绍，然后定义一个邻接矩阵结构体来储存边的信息，最后定义一个无向图类型的结构体来储存顶点的信息，边的信息，顶点的个数，边的条数。

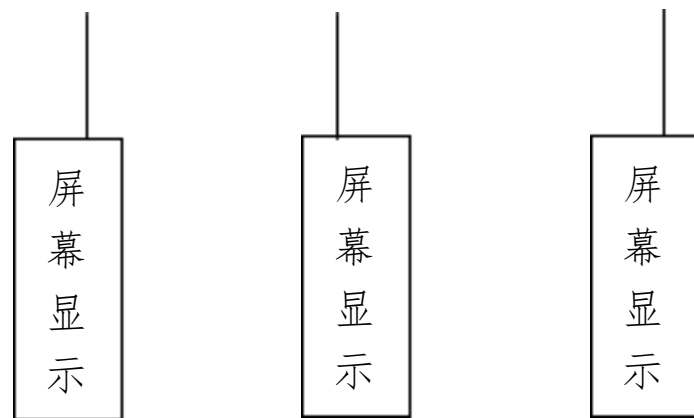
最后游客按照显示屏上的提示来进行相关的操作。

2 概要设计

2.1 程序的模块图

本软件的算法依据无向图的操作通过查找函数查找景点的信息，通过费洛伊德函数来查找最短距离，开始时首先从文件当中读取景点的编号、名称、介绍和两个景点之间的距离即权值，然后将其加入到图当中，再调用查找函数查找景点的信息，调用费洛伊德函数来查找最短距离，调用退出函数实现退出功能，其模块图如图 2.5 所示：





2.5 模块图

2.2 主函数的概要设计

基于程序的操作要求，对于主函数的设计首先是显示一个可视化的操作界面提醒游客进行相关的操作和提示游客其可供选择的景点的名称，便于其在后面的操作过程当中能够快速方便的找到其需要查找的景点的名称。而后就是一个 `switch ()`；的选择函数，提供查找景点信息，查找两个景点之间的最短距离和退出的相关的选择操作而后进入到每一个操作界面当中，从而实现所需要的功能。

2.3 查找介绍函数的概要设计

当游客选择了要查找景点的信息的介绍这一项功能的时候，就会进入到查找的界面，对于查找景点信息就是利用 `Search()`；函数，当游客输入景点的名称的时候看其是否与文件当中的数据相匹配，如果有则输出它的介绍，如果没有则输出错误的提示提醒游客进行相关的操作来进入到正确的操作过程当中。

2.4 查找最短路径函数的概要设计

对于查找最短路径的这一项功能，可以利用迪杰斯特拉算法，但我是用的费洛伊德算法，相对来说步骤跟简单一点。后面有详细介绍。

先手稿绘制所有景点的分布，利用 `printf()`；函数打印分布图的框架构造。各景点之间用线条链接，通过分布图能全面的对校园各景点有个方位感。

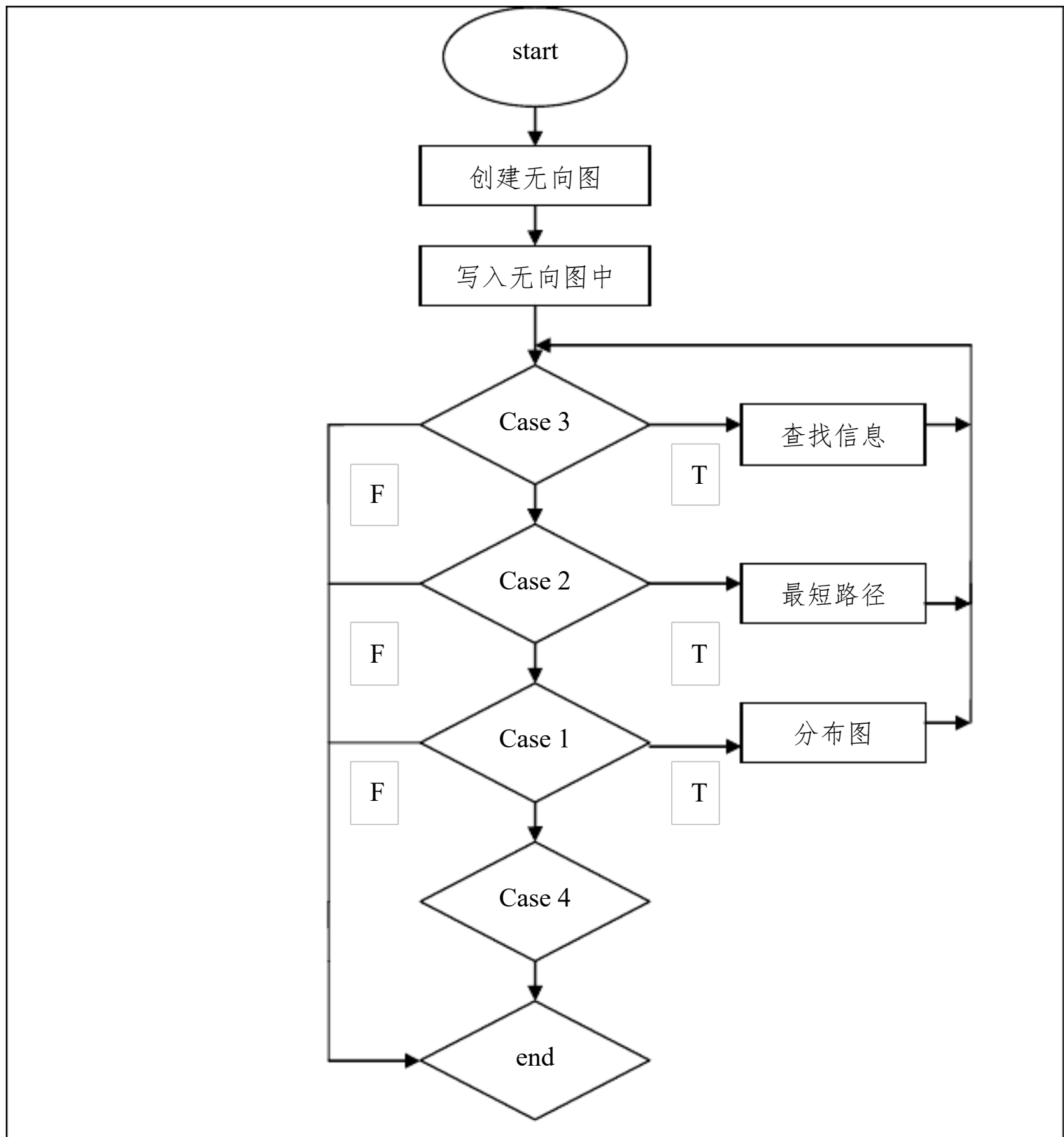
2.6 退出函数的概要设计

关于退出函数，则是当游客执行完了他想要进行的操作过后选择退出的功能的时候就调用退出函数 `exit(0)`；跳入到退出界面实现退出的功能。

3 详细设计

3.1 程序的流程图

当我们想要更加实际的了解一个程序的算法过程的时候，我们就要依据程序的流程图来给我们一个比较实际的过程，从流程图当中能够更加清楚整个程序实现的过程是怎样的。其流程图如图 3.1 所示：



3.1 流程图

3.2 主函数的详细设计

主函数是一个程序的主体，当我们要进行我们所需要的操作的时候我们就要根据主函数中的显示信息和它给我们的相关的提示信息来进行所需要的操作，因此在这次的程序实现的过程当中，调用 `Browser()`；提示游客根据 `switch()`；的选择语句，选择来进入到相关的操作界面实现程序的基本功能。

当游客选择了要查找景点的信息的介绍这一项功能的时候，程序就会调用 Search()函数进入到查找景点的介绍的界面，当游客输入了需要查找的景点的编号的时候，程序通过结构体，自动匹配相应的信息，查找是否有这个景点。

```
void Search(MGraph *G)
```

```
{
```

```
    int k,flag=1;
```

```
    while(flag){
```

```
        请输入要查询的景点编号
```

```
        if(k<=0||k>G->vexnum){
```

```
            景点编号不存在!请重新输入景点编号
```

```
        }
```

```
        if(k>0&&k<=G->vexnum)    flag=0;
```

```
    }
```

编 号	景 点 名 称	简 介
%-4d	%-16s	%-62s

},找到将它的编号返回，并输出它的介绍，没有找到这输出错误提示，提醒游客进行相关的操作进入正确的操作过程当中。

3.4 查找最短路径函数的详细设计

当游客选择了要查找两个景点之间的最短距离这一项功能的时候，函数进入到查找两个景点之间的最短距离的操作界面当中，当游客输入了两个景点的名称过后，程序会判断是否有这两个景点，如果有则返回他们各自的编号，并调用 Floyd();函数进入到查找最短路径问题的程序当中。

```
void Floyd(MGraph *G)
```

```

int v,u,i,w,k,j,flag=1,p[14][14][14],D[14][14];

for(v=1;v<=G->vexnum;v++)

    for(w=1;w<=G->vexnum;w++)

        {

            D[v][w]=G->arcs[v][w].adj;

            for(u=1;u<=G->vexnum;u++)

                p[v][w][u]=0;

            if(D[v][w]<INFINITY)

                {

                    p[v][w][v]=1;p[v][w][w]=1;

                }

        }

for(u=1;u<=G->vexnum;u++)

    for(v=1;v<=G->vexnum;v++)

        for(w=1;w<=G->vexnum;w++)

            if(D[v][u]+D[u][w]<D[v][w])

                {

                    D[v][w]=D[v][u]+D[u][w];

                    for(i=1;i<=G->vexnum;i++)

                        p[v][w][i]=p[v][u][i] || p[u][w][i];

                }

```

```
while(flag)
```

```
{
```

```
if(k<=0 || k>G->vexnum || j<=0 || j>G->vexnum)
```

```
{
```

景点编号不存在!请重新输入出发点和目的地的编号

```
}
```

```
if(k==j)
```

```
{
```

出发点和目的地一样!请重新输入出发点和目的地的编号

```
}
```

```
if(k>0 && k<=G->vexnum && j>0 && j<=G->vexnum)
```

```
flag=0;
```

```
}
```

最短游览路线:

```
if(k>j){
```

```
for(u=G->vexnum;u>0;u--)
```

```
if(p[k][j][u] && k!=u && j!=u)
```

```

if(k<j){

    for(u=1;u<=G->vexnum;u++)

        if(p[k][j][u] && k!=u && j!=u)

            总路线长

}

```

已知有 n 个顶点的有向图,佛洛伊德算法可以求解出每一对顶点之间的最短路径。假设使用邻接矩阵 $d(i, j)$ 来对图进行存储, $d(i, j)$ 表示 u 到 v 之间的距离, 但是该距离不一定是最短距离。佛洛伊德算法的基本思想是:为求顶点 $u \rightarrow v$ 之间的最短距离,需要进行 n 次试探。首先将 u 加入路径,考虑路径 $u \rightarrow 0 \rightarrow v$ 是否存在,如果存在,则比较 $u \rightarrow v$ 和 $u \rightarrow 0 \rightarrow v$ 的路径长度,取长度短的路径作为 $u \rightarrow v$ 的路径,记作 (u, v) 。接着在路径上再增加一个顶点 u ,比较 $u \rightarrow v \rightarrow v$ 和 (u, v) 的路径长度,取长度短的路径作为 (u, v) 。不断将顶点 $u, v, \dots, n-1$ 加入进行试探,最后得到的 (u, v) 必定为 $u \rightarrow v$ 的最短路径。若使用数组 $dk(i, j)$ 表示加入顶点 k 后,最短路径长度的变化情况,使用数组 $pk(i, j)$ 表示加入顶点 k 后,最短路径上顶点的变化情况,这样就求得了最短路径和最短路径长度。

3.5 景点分布图的详细设计

这里不详细介绍,具体代码,查看附录 `browse_view_distribute()` 函数。

3.6 退出函数的详细设计

对于退出函数，当游客选择了退出这一个操作的时候，程序就会调用 `exit(0)`；函数实现退出主函数的功能。最后会提示游客，欢迎下次继续使用！

3.7 数据结构的详细设计

本软件的数据结构包括 3 个部分：

1. 邻接矩阵

```
typedef int AdjMatrix[MAX_VERTEX_NUM][MAX_VERTEX_NUM];
```

定义一个邻接矩阵，用邻接矩阵来定义和储存边的相关信息。

2. 顶点的结构体

```
typedef struct Vertex// 定义图中顶点的数据类型
```

```
{
```

```
int num;// 景点编号
```

```
char name[30];// 景点名称
```

```
char introduction[200];// 景点介绍
```

```
}Vertex;
```

定义一个顶点的结构体，用来储存景点的编号、景点得名称和景点的介绍等关于景点的信息。

3. 无向图的结构体

```
typedef struct // 定义图的数据类型
```

```
{
```

```
Vertex vexs[MAX_VERTEX_NUM];// 顶点的结构体
```

```
AdjMatrix arcs;// 边的邻接矩阵
```

```
int vexnum,arcnum;// 顶点的个数， 边的个数
```

```
}MGraph;
```

定义一个图的结构体，用来储存顶点的信息、边的信息、顶点的个数和边的个数等相关的信息便于我们以后在用的时候能够方便快捷的调用。

定义好这些结构体后，当我们以后需要调用的时候，我们就能够方便快捷的调用这些结构体，从而使得我们在运行程序的时候能够更加的快速能够提高我们的程序的运行效率，大大的节省了我们的时间还使得程序变得更加的简单。

4 软件测试

4.1 菜单的测试

对于菜单函数的测试，首先菜单是一个可视化的界面，它能够提示游客依据显示屏上出现的提示来进行相关的操作，查看所有的景点从而方便游客进行相关的操作，因而在运行程序的时候首先就会进入到菜单函数当中，经过测试其能够实现我们所要实现得基本功能，其效果图如图 4.1 所示：

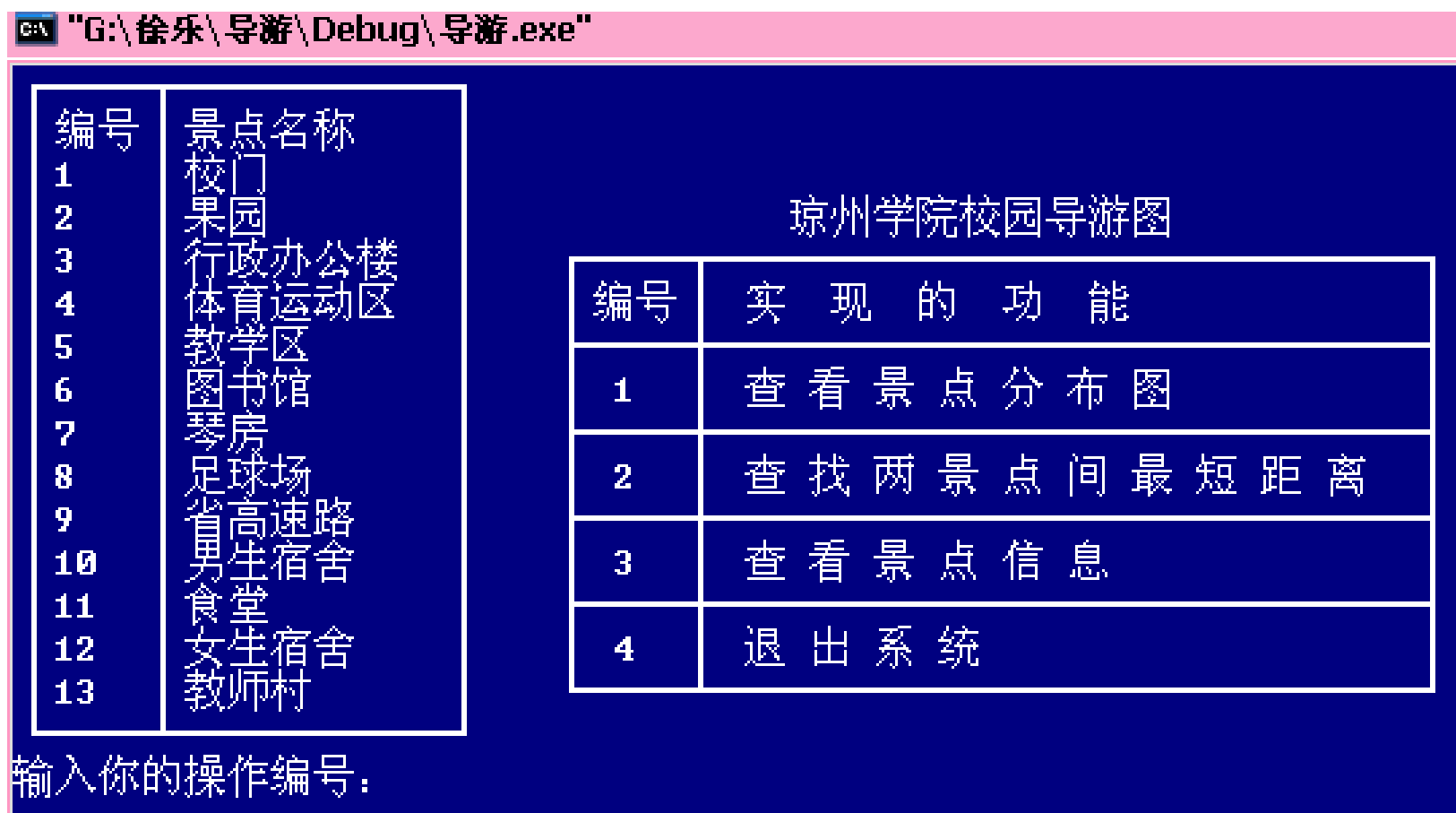


图 4.1 菜单

4.2 查找景点简介的测试

对于查找景点的介绍的测试，首先依据显示屏上的提示首先输入要进行的操作输入 3 进入查找景点信息的操作界面，然后输入需要查找的景点的名称即可显示出景点的介绍信息，经过测试可以得出其没有什么错误，程序能够按照我的要求实现它的功能，其效果图如图 4.2 所示：

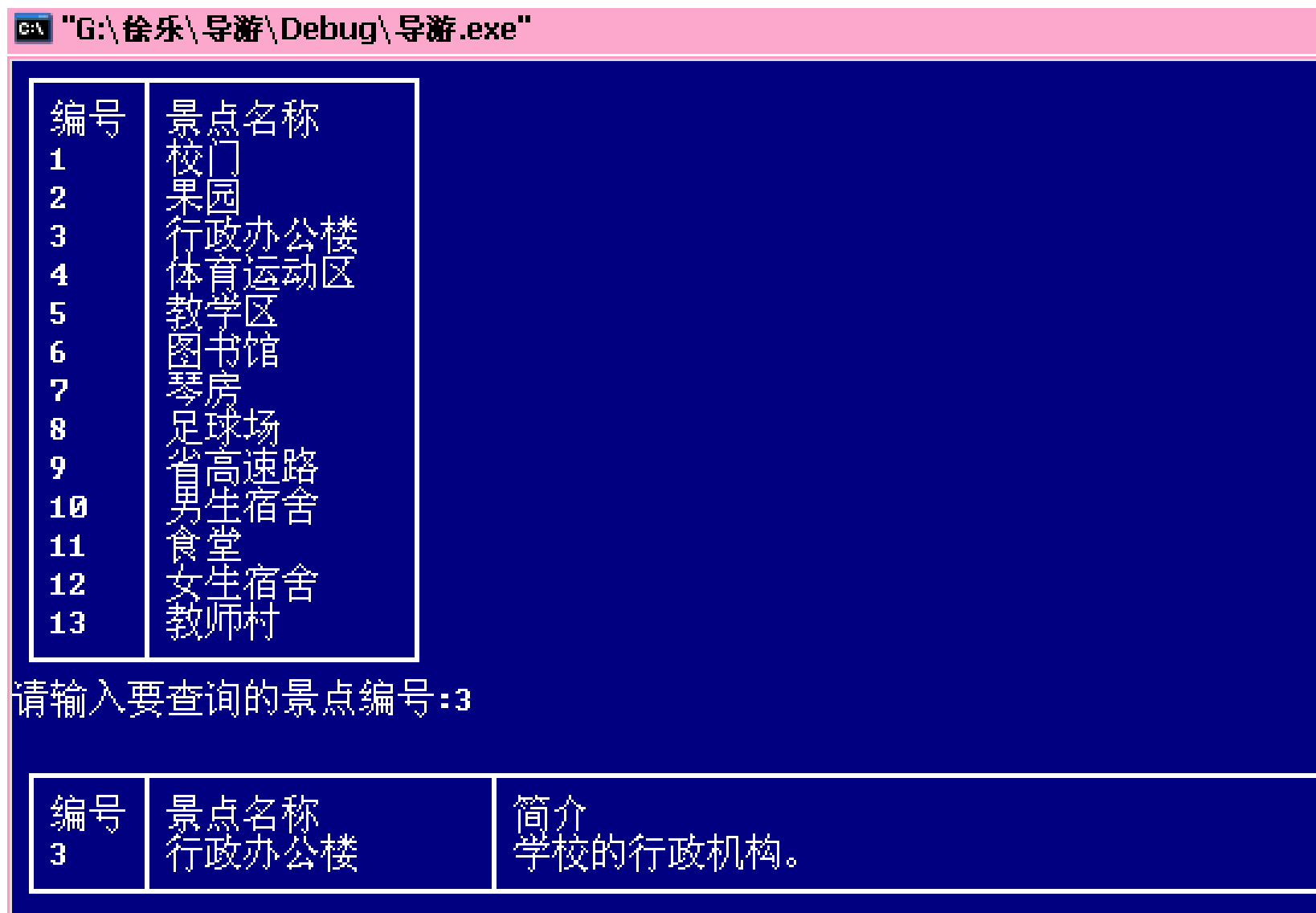


图 4.2 查找景点信息

4.3 查找两个景点之间的最短距离的测试

同查找景点的信息一样，对于查找景点之间的最短距离的测试，我们就要依据提示输入 2 进入到查询最短路径的界面，依次输入所需要查找的两个景点就会显示出怎样到达这两个景点并显示出它们之间的最短路径，经过测试可见程序能够按照我的要求来实现其所需要的功能，其效果图如图 4.3 所示：

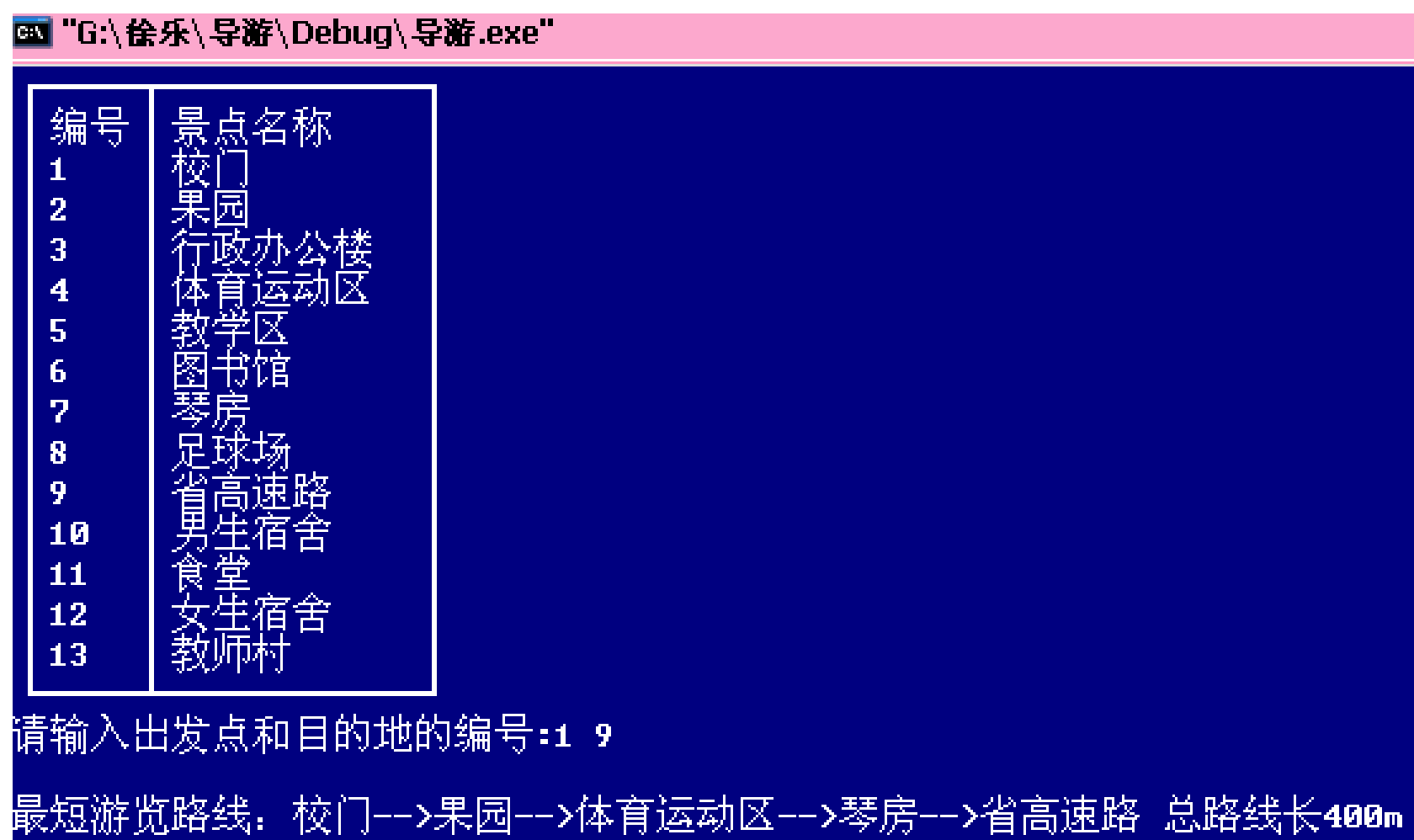


图 4.3 查找两个景点之间的最短距离

4.4 查看景点分布图的测试

对于查看景点分布图的测试，我需要依据显示屏上的提示，需要输入 1 进入到分布图的界面，系统就会直接调用 `browse_view_distribute()` 函数，在屏幕上打印出景点的分布图。

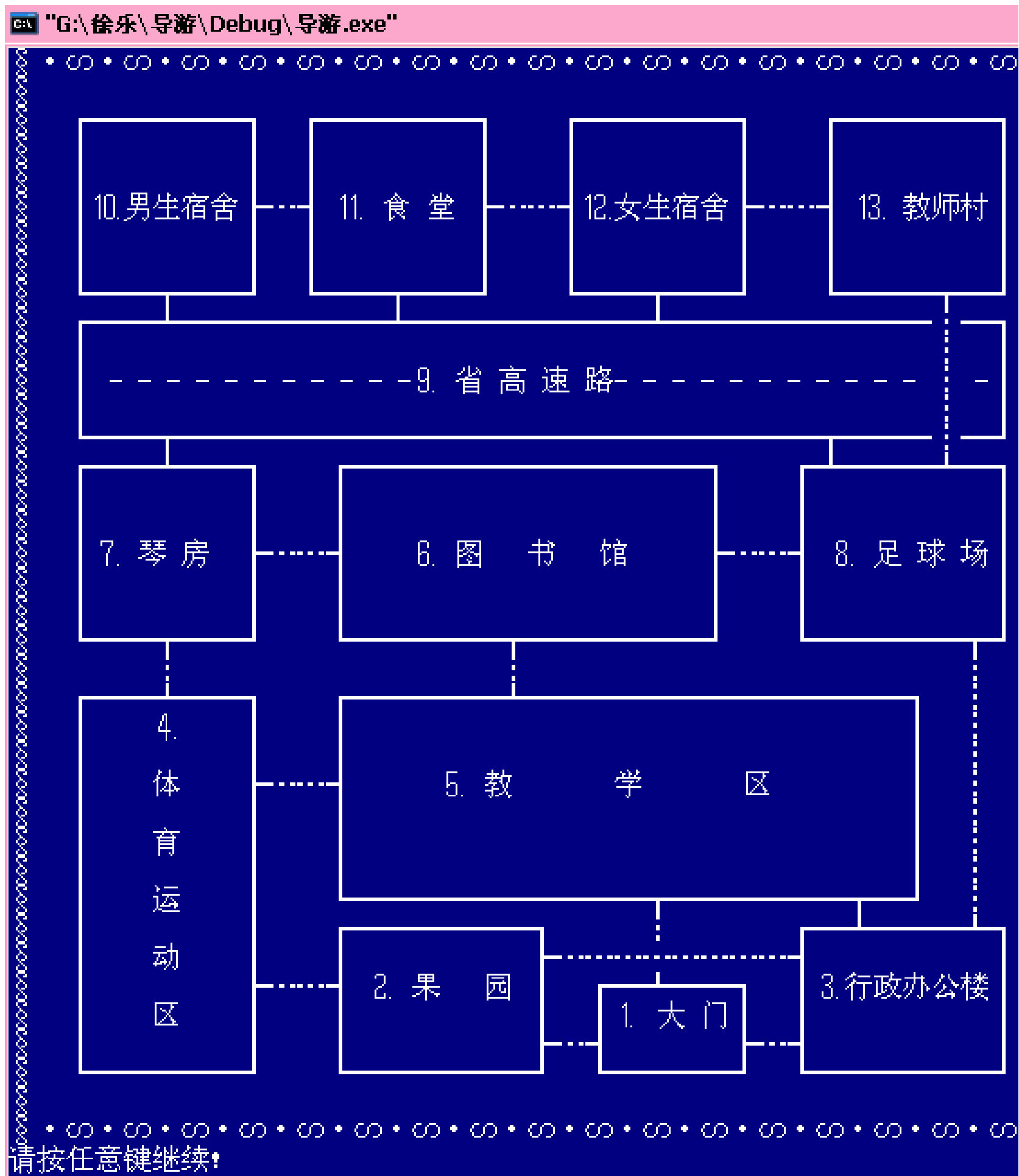


图 4.4 景点分布图界面

4.5 退出的测试

原理同上，对于退出函数的测试，我需要依据显示屏上的提示，需要输入 4

进入到退出的界面，系统就会直接调用退出的函数，显示出“欢迎下次继续使用”的话，退出了系统，其效果图如图 4.4 所示：

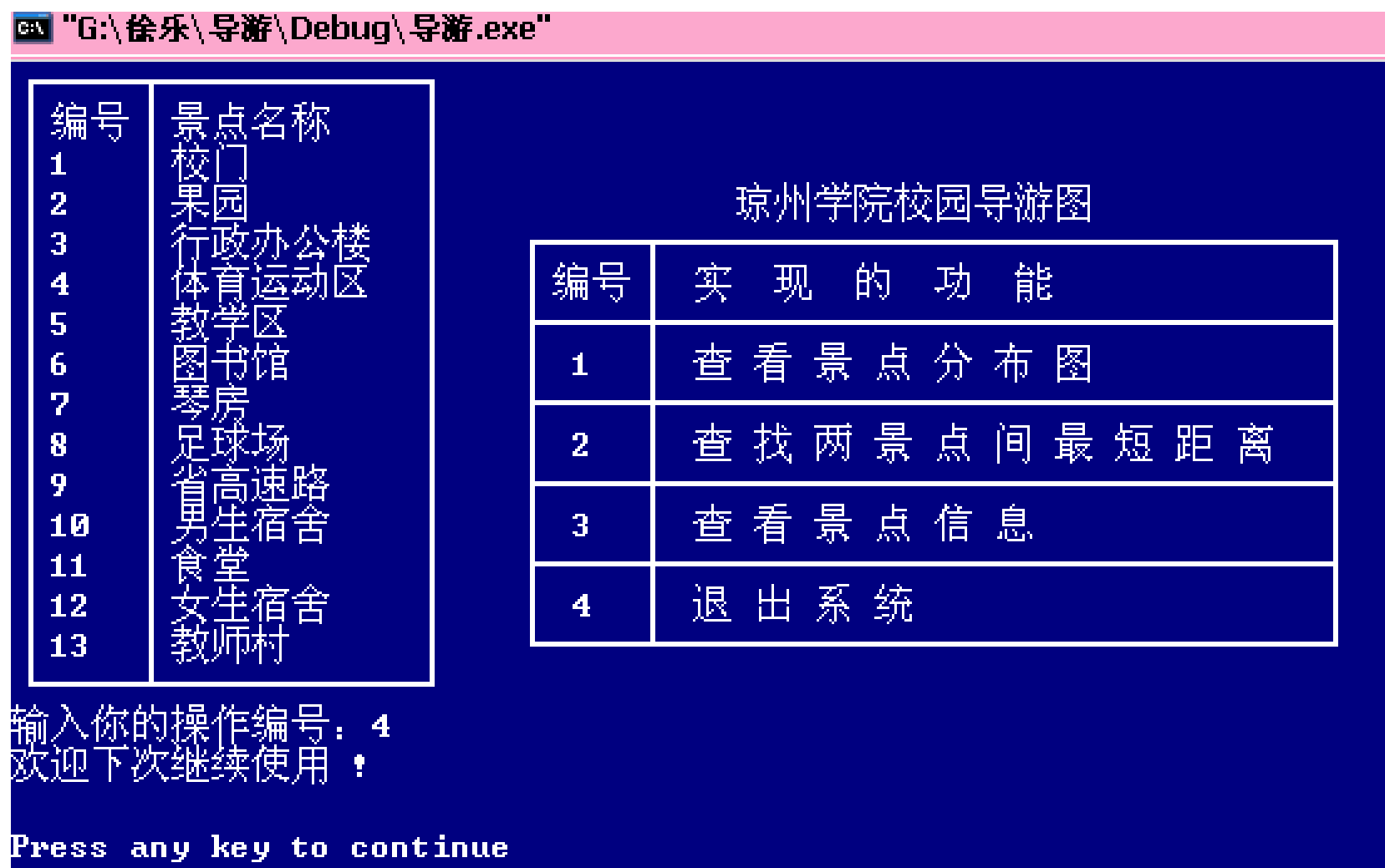


图 4.5 退出界面

5 软件使用说明

对于软件的使用，对于第一次使用软件的游客来说，要让他们在第一次用的时候就能够快速轻松的掌握软件的使用，因此在程序一开始运行的时候，我们要进行如下的操作：

(1) 首先我会给游客提供一个可视化的菜单操作界面，在显示屏上提示用户其可以进行的操作和他能够查询的景点的编号、名称。

(2) 用户输入了“1”后，屏幕上会显示出所有景点的位置关系平面图。能够大致了解景点的分布。

(3) 当用户输入了“2”后，进入到查询任意两景点间最短路径的界面，然后提示用户依次输入两个景点的编号，程序就会将这两个景点的最短路径给我们表示出来并显示出最短路径是多少。

(4) 用户输入了“3”后，进入到查询景点信息的界面，然后提示用户输入景点的编号（一次限一个），程序就会显示出这个景点的详细介绍。

(5) 当用户输入了“4”后，进入到退出界面，这时系统就会显示“欢迎下次继续使用！”的提示语，最后按下任意键退出系统。

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/026104111030010230>