算法与数据结构实践》

课程报告

**题 目： 地图导航查询系统（最短路径问题）**

**学 号： 201941301413**

**姓 名： 黄清贤**

**专业班级： 2019级机器人4班**

**起止时间： 2020年9月-2020年11月**

**报告批阅成绩：**

目录

**系统需求分析**1

地图导航查询系统介绍2

地图导航查询系统用户说明……………………………………………………………...….2

地图导航系统功能介绍2

地图导航查询系统具体功能需求分析………………………………………………………3

**系统总体设计**4

系统结构设计4

数据结构设计4

邻接表设计………………………………………………………………….…………...4

顶点表设计………………………………………………………………….…………….4

图设计………………………………………………………………….…………………4

**系统详细设计与分析**

文件操作功能详细设计………………………………………………………………………5

创建图功能……………………………………………………………………………………5

添加点功能……………………………………………………………………………………5

添加边功能……………………………………………………………………………………5

删除点功能……………………………………………………………………………………..5

删除边功能……………………………………………………………………………………..5

改变点功能……………………………………………………………………………………..5

改变边功能……………………………………………………………………………………..6

求两点最短距离功能…………………………………………………………………………...6

输出地图信息功能…………………………………………………………………………...6

界面层设计……………………………………………………………………………………6

**系统实现与测试**……………………………………………………………………………………..7

**总结**…………………………………………………………………………………….…………10

# 1系统需求分析

1.1**地图导航查询系统介绍**

随着时代不断发展，为了方便人们能够更好的对地图上两点之间的最短距离有更好的把控，以及实现添加地点、添加路径等等地图操作，使得地图的信息愈发丰富。本地图导航查询体统应运而生。本系统要求系统界面友好、使用简单、提供添加点、添加边、删除点、删除边、改变点、改变边、最短路径问题等全面地图导航功能。

1.2**地图导航查询系统用户说明**

本系统的用户能够对地图实现多个操作例如添加点、添加边等。并且能够使得地图信息永久保存。

1.3**地图导航系统功能介绍**

对点进行操作

对边进行操作

地图导航查询系统

修改边

删除边

添加边

修改点

删除点

添

加

点

信息保存与

输出

最短路径问题

系统主要功能包括：对点进行操作、对边进行操作、最短路径问题、信息保存与输出模块，具体说明如下：

1. 对点进行操作：主要包括添加点、删除点、修改点的操作。
2. 对边进行操作：主要包括添加边、删除边、修改边的操作
3. 最短路径问题：主要实现地图上两点的最短距离的问题
4. 信息保存与输出：将地图中的信息永久保存于文件中，输出则是大概描绘地图。

1.5 地图导航查询系统具体功能需求分析

功能介绍：用户打开系统后，系统会从文件中读取当前地图信息，而后用户可以选择多个操作进行。

1. 输入

用户输入所需功能

1. 输出

对应的结果

# 2系统总体设计

本系统是桌面应用程序，一共分成三层，如下图所示。

**界面层**

**业务功能层**

**数据结构层**

**持久化数据层**

**2.2数据结构设计**

**2.2.1邻接表设计：指向顶点位置、下一条弧的指针、权值**

typedef struct ArcNode { //

int index; //该弧指向顶点的位置

struct ArcNode\* nextarc; //指向下一条弧的指针

int weight; //网的权值

}ArcNode;

**2.2.2顶点表设计：数据与、指向第一个表结点的地址**

typedef struct VNode {

char data[20]; //数据域

ArcNode\* firstarc; //指向第一个表结点的地址

}VNode,AdjList[100];

**2.2.3图设计：顶点表数组，图的顶点数和弧数**

typedef struct ALGraph{

AdjList vertices; //顶点表数组

int vexnum, arcnum; //图的当前顶点数和弧数

}ALGraph;

**3 系统详细设计与实现**

**3.1 文件操作功能详细设计**

开始->定义 \*fp->以只读方式打开文件->判断文件是否读取完毕。若未读取完毕，则将文件数据依次输入。在读取完毕后关闭文件。

在用户完成所需要求之后，以重写方式打开文件，将数据依次输入至文件中，而后保存后关闭文件。

**3.2 创建图功能**

void createALGraph(ALGraph\* G, FILE\* fp)

从文件读取图的顶点数、边数，以及每个顶点的数据，相连两点的下标以及其距离。对图进行操作将数据保存，即可创建图。

**3．3添加点功能**

void insertVex(ALGraph\* G)

用户输入所加点的信息，将图的顶点数加1，然后将顶点信息保存至新加顶点坐标下。并将其顶点表的下一条弧指针设为NULL，即可完成创建点功能。

**3.4添加边功能**

void insertArc(ALGraph\* G)

用户输入所加边的两顶点的下标以及其权值，在进行输入格式检查后，由下标找出该点的邻接链表，将链表遍历至最后一点，将另一个点的信息以及权值添加至链表末尾，对两个点进行操作后即可连接两点。

**3.5删除点功能**

void delVex(ALGraph\* G)

用户输入所删除点的下标，在进行输入格式检查后，首先找出与需删除点的下标，然后对该点的邻接链表进行遍历，找到需删除的点，在链表中将其信息删去（即node2 = node2->nextarc）。

将与所删除点与其所有连接的点的边删去后，图的边数减去相应的数目，图的顶点数减一。另外将原本下标大于所删除点下标的点的下标减1。

**3.6删除边功能**

void delArc(ALGraph\* G)

用户输入所删除的边的两点的下标，在进行输入格式检查后，对两点的邻接链表进行遍历找出另一点的下标，分情况将该点删去。最后将图的边数-1，即可实现删除边功能。

**3.7改变点功能**

void changeVex(ALGraph\* G)

用户输入所改点的下标，在进行输入格式检查后，输入所改成的信息，将该信息修改即可实现改变点功能。

**3.8改变边功能**

void changeArc(ALGraph\* G)

用户输入所改边的两顶点的下标，在进行输入格式检查后，分别对两点的邻接链表进行遍历以找到另一顶点的信息，修改权值为用户所输入权值即可实现改变边功能。

**3.9求两点最短距离功能**

int lowestWeightSum(ALGraph\* G)

应用dijkstra算法，设置两个数组sorted[]、waited[]，分别表示已找出最短距离和待找出最短距离的顶点。给数组赋初值INF（int类型最大值）。用户输入所求两点后，从其中一点开始。查找当前所能到达点中的最小值，由于该网没有负值，则当前所能找到的最小值即为到达该点的最小值，将其存入sorted数组中。而后将该点所连接的点都设为可到达的点，找出当前所能到的点的距离最小值。设置for循环，每次循环找到到达一个点的最小距离，则循环G->vexnum次，即可找出该点到网上每一点的最小距离。最后返回sorted[j]，j为用户输入的另一下标。即可实现两点最短距离功能。

**3.10输出地图信息功能**

void output(ALGraph\* G)

双重循环，遍历每一点及每一点的邻接链表，输出相关信息，即可实现输出地图信息功能。

**3.11界面层设计**

本系统采用c语言开发，界面采用字符界面方式，要设计字符菜单来让用户选择相应的功能，信息的显示和输入都采用字符界面方式。系统主界面设计如下：

|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|

1.添加点 2.添加边 3.删除点

4.删除边 5.修改点 6.修改边

7.最短距离 8.输出 0.保存并退出

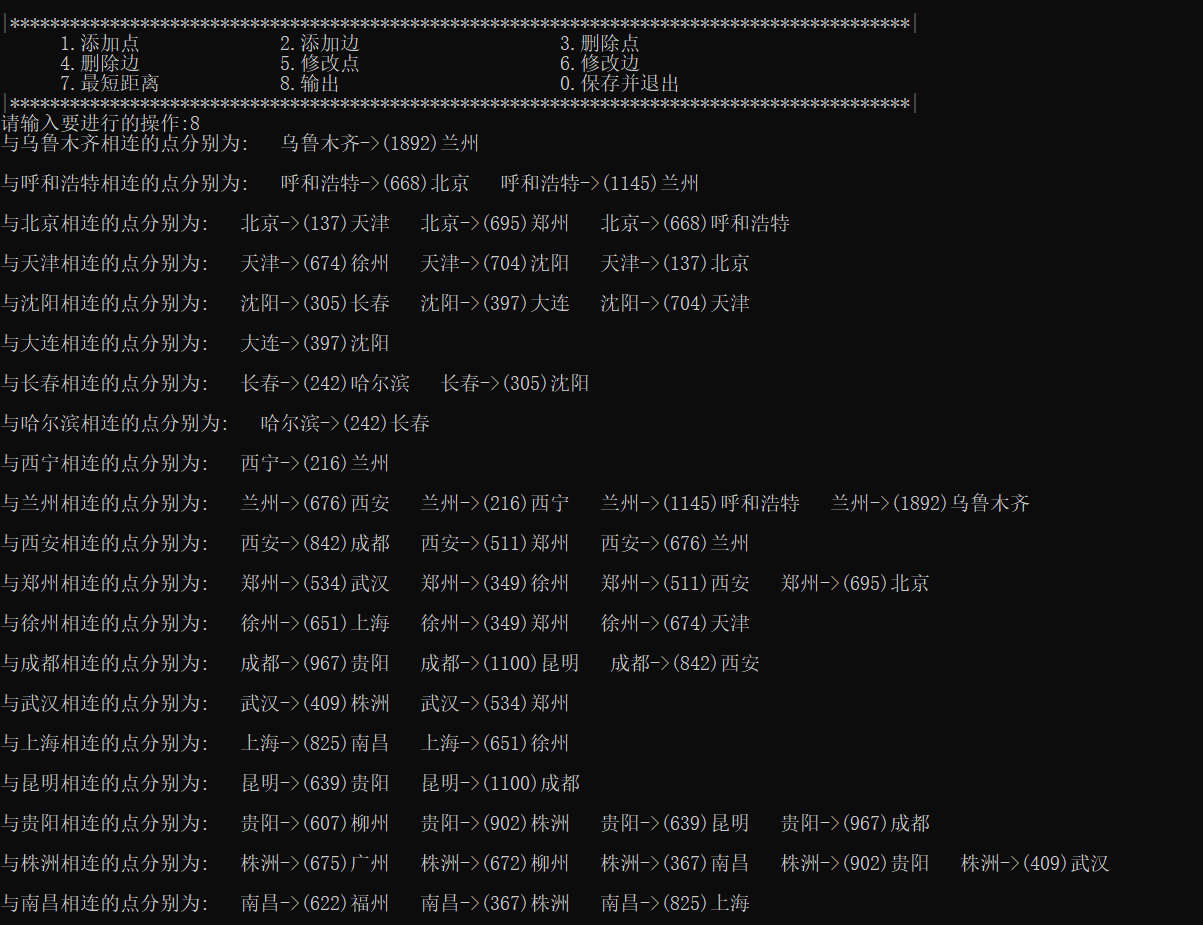
|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|

4系统实现与测试

系统实现，系统测试，并对结果进行分析说明，建议设计测试用例。

4.1数据读取测试

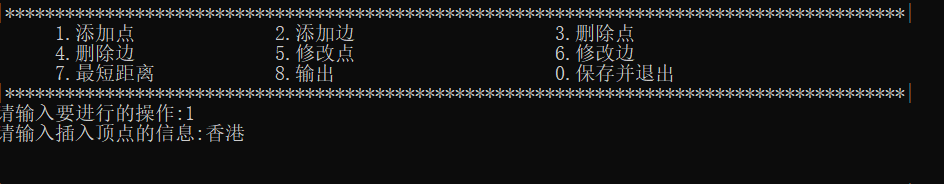
进行输出功能

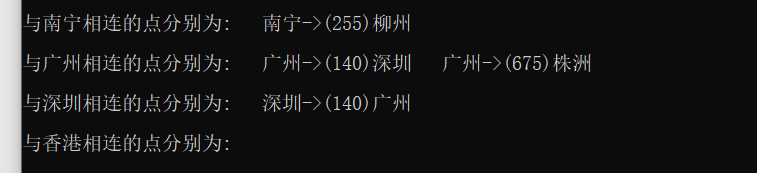


测试结果：通过

**4.2 对点进行操作**

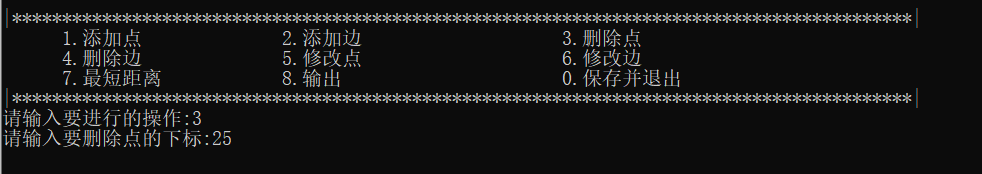
**4.2.1添加点功能：**

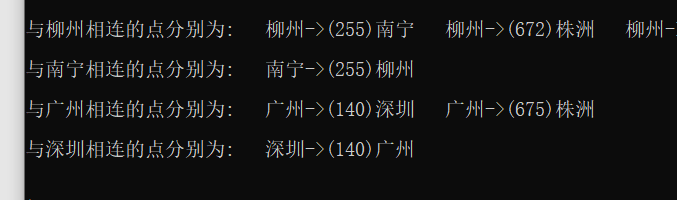


**输出**

**测试结果：通过**

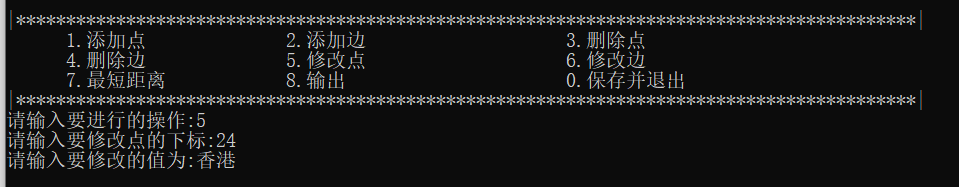
**4.2.2删除点功能**



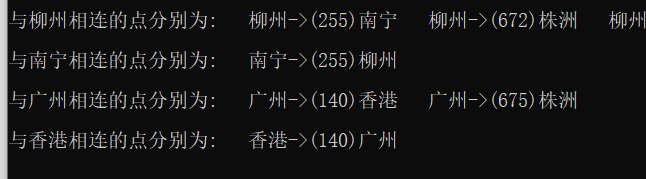
**输出**

**输出结果：通过（香港已被删去）**

**4.2.3修改点功能**



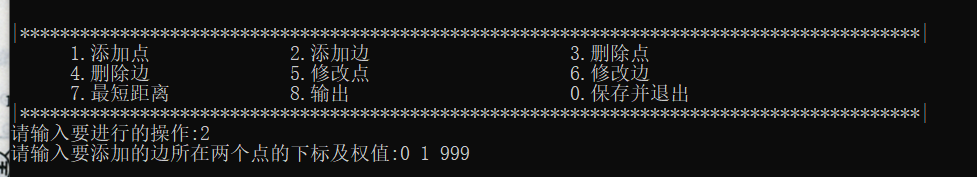
**输出**



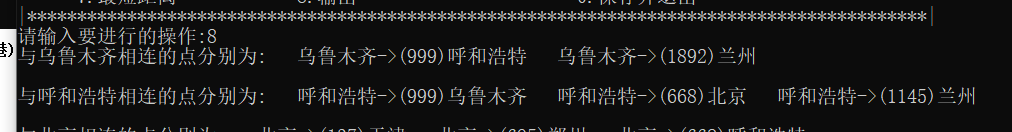
**输出结果：通过（将深圳改为香港）**

**4.3对边进行操作**

**4.3.1添加边功能**

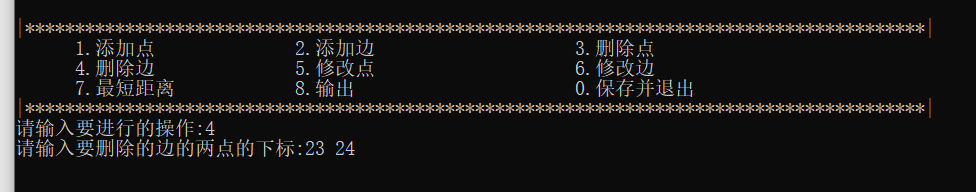


**输出**

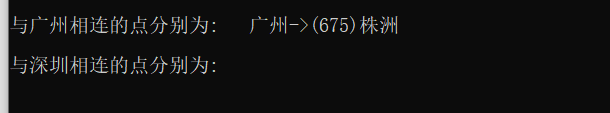


**输出结果：通过（将乌鲁木齐与呼和浩特相连，距离为999）**

**4.3.2删除边功能**

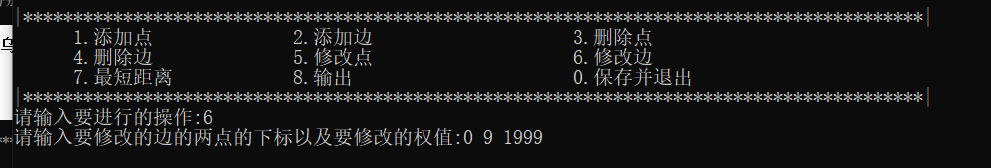


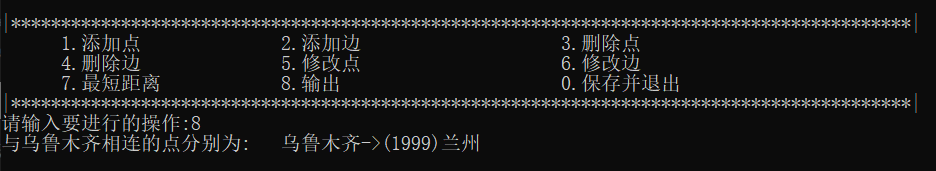
**输出**



**输出结果：通过（将广州与深圳之间的边删去）**

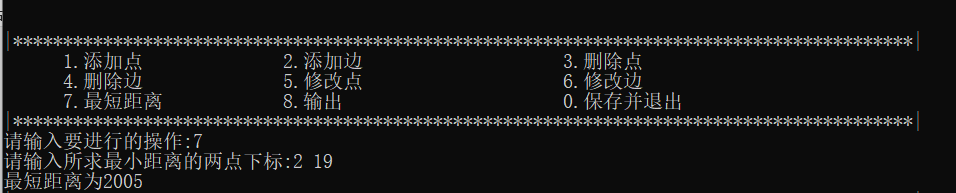
**4.3.3修改边功能**



**输出**

**输出结果：通过（将乌鲁木齐与兰州的距离由1892改为1999）**

**4.4最短距离功能**



**输出结果：通过（从北京到南昌最近距离为北京->(695)郑州->(534)武汉->(409)株洲->(367)南昌，距离为695+534+409+367=2005）**

总结

问题考虑不周全，导致在写的过程中会顾此失彼，bug不断。

此项目让我熟悉了图的结构以及对图的许多操作。