

HBU校园导航实验报告

韩旭

韩旭

目录

[数据结构大作业]HBU Guide	1
河北大学校园导航	1
【问题描述】	1
【需求分析】	1
实现提示：一般情况下，校园道路是双向通行的，可设计校园平面图是一个无向图。顶点和边均含有相关信息。选做内容：（1）提供图的编辑功能：增删景点；增删道路；修改已有信息等。（2）校园导游图的仿真界面。	2
TODO	3
函数列表	3
数据结构实验报告	4
【概要设计】	4
1. 抽象数据类型定义：	4
2 主要功能模块	5
3 主模块流程	6
【详细设计】	6
【调试分析】	7
【使用说明书及测试结果】	8

韩旭051

[数据结构大作业]HBU Guide

河北大学校园导航

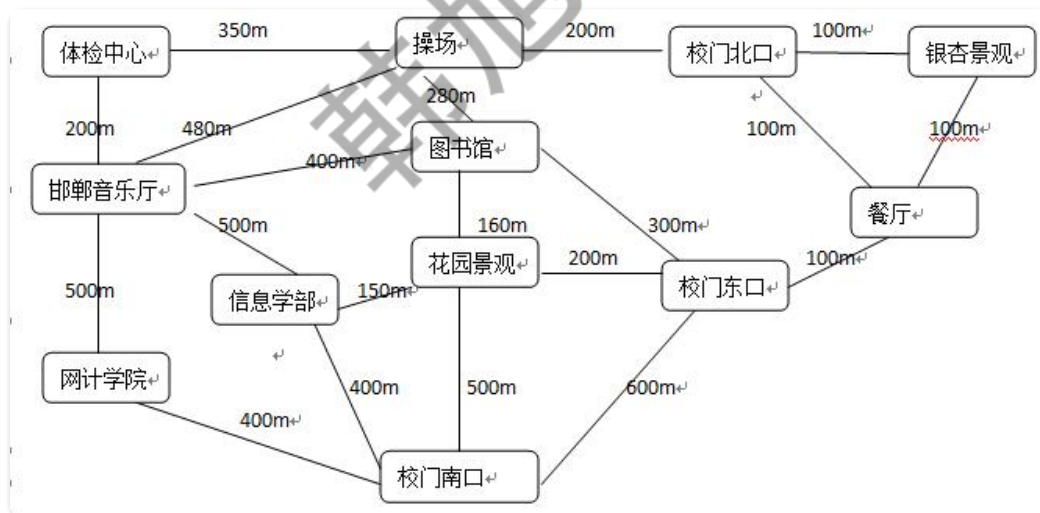
校园导航问题

【问题描述】

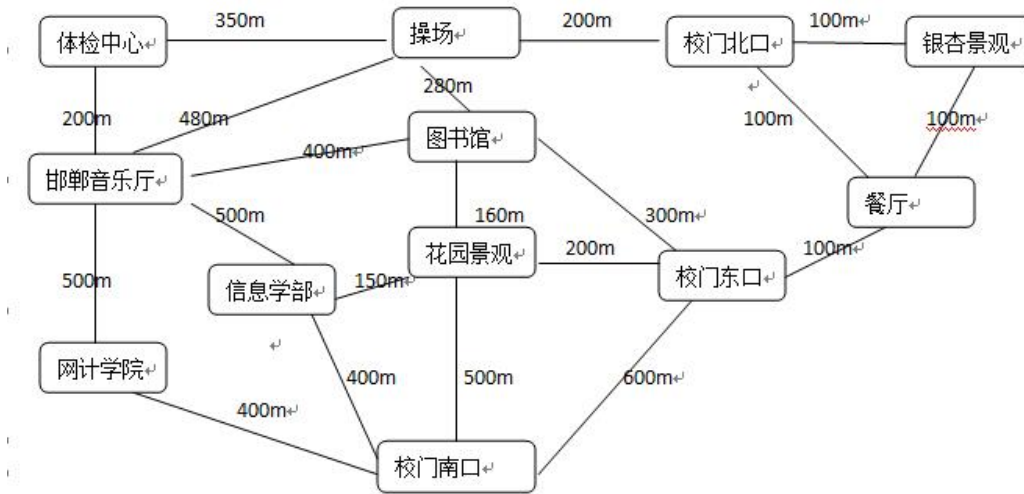
以我校为例，设计一个校园导航系统，主要为来访的客人提供信息查询。系统有两类登录账号，一类是游客，使用该系统方便校内路线查询；一类是管理员，可以使用该系统查询校内路线，可对校园景点路线可编辑。

【需求分析】

设计学校的平面图，至少包括10个以上景点（场所），每两个景点间可以有不同道路，且路长也可能不同，找出在游人所在景点到其他景点的最短路径，或游人输入的任意两个景点的最短路径。要求：（1）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，路径权重为路径长度。（2）为游人提供任意景点相关信息查询。（3）为游人提供任意景点的问路查询，即任意两个景点之间的最短路径。



text



实现提示：一般情况下，校园道路是双向通行的，可设计校园平面图是一个无向图。顶点和边均含有相关信息。选做内容：（1）提供图的编辑功能：增删景点；增删道路；修改已有信息等。（2）校园导游图的仿真界面。

数据结构校园导航

抽象数据类型定义

(1) 景点 顶点名称 代号 顶点信息简介 —— `Typedef struct{ Int num; Char name[100]; Char features[200]; }`

(2) 图的存储结构: —— `Typedef int EdgeType; Typedef struct{ VertexType vexs[MaxVertexNum]; EdgeType edges[MaxVertexNum][MaxVertexNum]; Int n, e; } MGraph;`

主要功能模块

- (1) 创建图的邻接矩阵存储结构 create()
- (2) 浏览图中任一景点介绍
- (3) 修改景点信息
- (4) 增加景点信息
- (5) 删除景点信息
- (6) 增加道路
- (7) 删除道路
- (8) 查找某一景点到其他景点的最短路径
- (9) 查找任一两个景点之间的最短路径。

主模块流程

- 管理员登陆 —— 可实现 (1) - (9) 功能操作
- 游客登陆 —— 在 (1) 基础实现基础之上, 可实现 (2) (8) (9) 功能操作

【详细设计】

1. 用C语言定义相关数据类型
2. 写出各模块伪代码算法
3. 画出函数间的调用关系图

【调试分析】

1. 调试中遇到的问题及对问题的解决方法
2. 算法时间复杂度和空间复杂度

【使用书说明及测试结果】

数据结构校园导航 三

抽象数据类型定义

- (1) 景点 顶点名称 代号 顶点信息简介 —— `Typedef struct{ Int num; Char name[100]; Char features[200]; }`
- (2) 图的存储结构: —— `Typedef int EdgeType; Typedef struct{ VertexType vexs[MaxVertexNum]; EdgeType edges[MaxVertexNum][MaxVertexNum]; Int n, e; } MGraph;`

主要功能模块

- (1) 创建图的邻接矩阵存储结构 create()
- (2) 浏览图中任一景点介绍
- (3) 修改景点信息
- (4) 增加景点信息
- (5) 删除景点信息
- (6) 增加道路
- (7) 删除道路
- (8) 查找某一景点到其他景点的最短路径
- (9) 查找任一两个景点之间的最短路径。

主模块流程

- 管理员登陆 —— 可实现 (1) - (9) 功能操作
- 游客登陆 —— 在 (1) 基础实现基础之上, 可实现 (2) (8) (9) 功能操作

【详细设计】

1. 用C语言定义相关数据类型
2. 写出各模块伪代码算法
3. 画出函数间的调用关系图

【调试分析】

1. 调试中遇到的问题及对问题的解决方法
2. 算法时间复杂度和空间复杂度

【使用说明书及测试结果】

text

TODO

- ☑1.构建函数列表 2019年11月30日21:36:15
- ☑2.构造存储数据结构体 2019年11月30日21:36:17
- ☑3.编写用户管理员登录界面, 以及函数调用。 2019年11月30日21:36:19
- ☑4.完善各个函数功能。

函数列表

- ☑//景点介绍


```
void introduct (void){
    return;
}
```
- ☑//查找游客所在景点与其他景点的距离


```
void Dijkstra(void){
    return;
```

```
}  
☑//查找游客指定的两个景点间的最短路径长度  
void Floyd(void){  
    return;  
}  
☑//修改景点信息  
void modifyInfo(void){  
    return;  
}  
☑//添加景点  
void addInfo(void){  
    return;  
}  
☑//删除景点  
void delInfo(void){  
    return;  
}  
☑//添加道路  
void addPath(void){  
    return;  
}  
☑//删除道路  
void delPath(void){  
    return;  
}  
☑//生成图  
void create(void){  
    return;  
}
```

数据结构实验报告

【概要设计】

1. 抽象数据类型定义：

(1) 景点 顶点名称 代号 顶点信息简介

```
Typedef struct{  
    Int num;  
    Char name[100];  
    Char features[200];  
} VertexType;
```

(2) 图的存储结构：

```

Typedef int EdgeType;
Typedef struct{
    VertexType vexs[MaxVertexNum];
    EdgeType edges[MaxVertexNum][MaxVertexNum];
    Int n, e;
} MGraph;

```

2 主要功能模块

- (1) 创建图的邻接矩阵存储结构 create()
- (2) 浏览图中任一景点介绍
- (3) 修改景点信息
- (4) 增加景点信息
- (5) 删除景点信息
- (6) 增加道路
- (7) 删除道路
- (8) 查找某一景点到其他景点的最短路径
- (9) 查找任一两个景点之间的最短路径。

```

//景点介绍
void introduct (void){
    return;
}
//查找游客所在景点与其他景点的距离
void Dijkstra(void){
    return;
}
//查找游客指定的两个景点间的最短路径长度
void Floyd(void){
    return;
}
//修改景点信息
void modifyInfo(void){
    return;
}
//添加景点
void addInfo(void){
    return;
}
//删除景点
void delInfo(void){
    return;
}
//添加道路
void addPath(void){
    return;
}
//删除道路
void delPath(void){
    return;
}
//生成图
void create(void){
    return;
}

```

3 主模块流程

管理员登陆，可实现（1）-（9）功能操作 游客登陆，在（1）基础实现基础之上，可实现（2）（8）（9）功能操作

```
printf("请输入您的密码: ");
char password[PASSWORDLENGTH];
scanf("%s", password);
getchar(); //一定要吃回车，不然影响下一次输入
if(strcmp(password, PASSWORD)==False) {
    printf("密码正确，即将进入系统\n");
    Administrator(); //转入超级管理员界面
}else{
    system("cls");
    printf("-----密码错误-----\n");
    printf("系统将在1s后回到主面板");
    Sleep(1000);
    system("cls");
}
break;
default:
printf("即将在1s后退出系统，感谢您的使用。");
Sleep(1000);
return 0;
```

【详细设计】

1. 用C语言定义相关数据类型

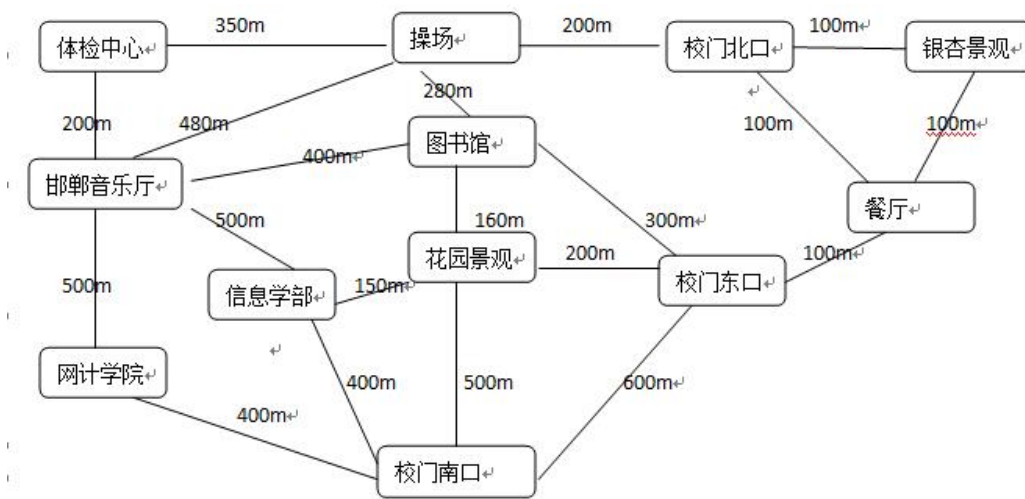
（1）景点 顶点名称 代号 顶点信息简介

```
Typedef struct{
    Int num;
    Char name[100];
    Char features[200];
} VertexType;
```

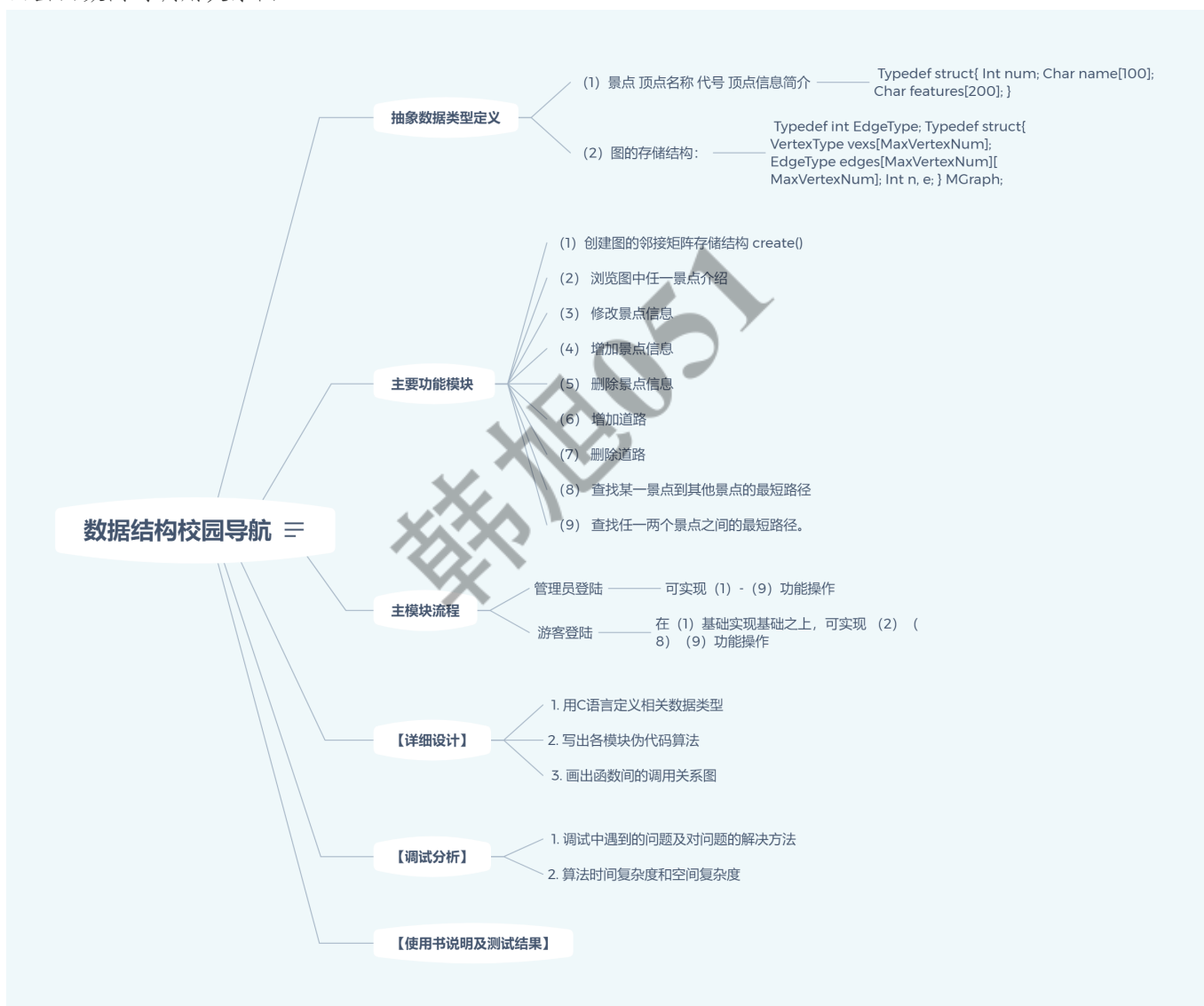
（2）图的存储结构：

```
Typedef int EdgeType;
Typedef struct{
    VertexType vexs[MaxVertexNum];
    EdgeType edges[MaxVertexNum][MaxVertexNum];
    Int n, e;
} MGraph;
```

2. 写出各模块伪代码算法



3. 画出函数间的调用关系图



【调试分析】

1. 调试中遇到的问题及对问题的解决方法

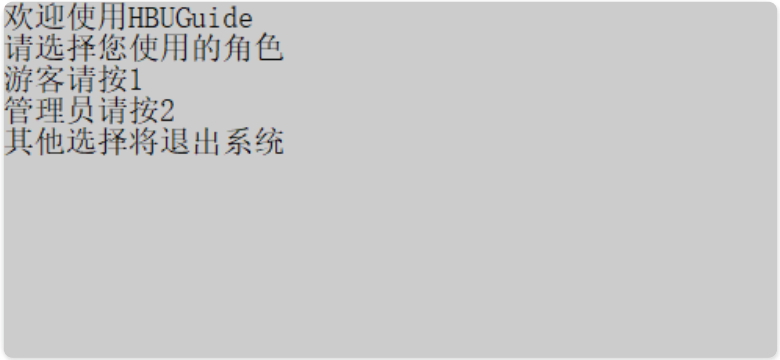
遇到算法查询最短路径问题，查找弗洛伊德算法和迪杰斯特拉算法通过搜索引擎，借鉴前人的思路和方法，进行学习和总结。

2. 算法时间复杂度和空间复杂度

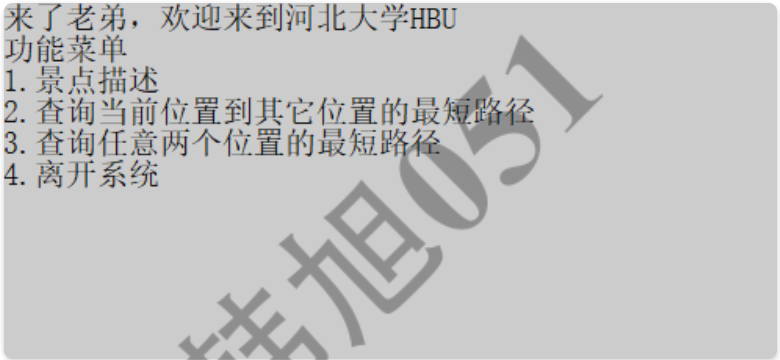
时间复杂度 $n * n$

空间复杂度 存储信息 $n * n$

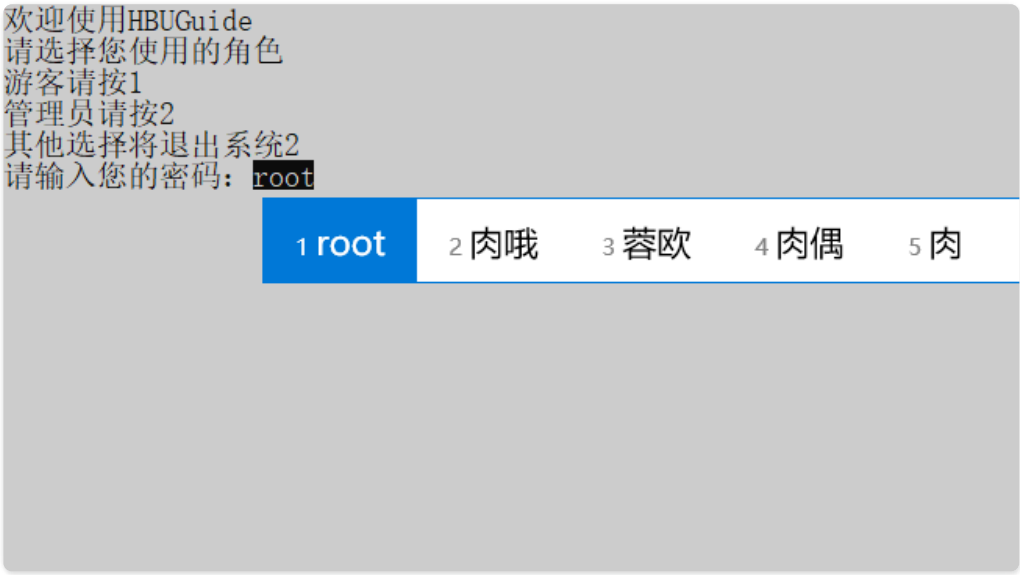
【使用说明书及测试结果】



登录界面



游客界面



管理员登录

只有管理员才能看到的管理系统

1. 景点介绍
2. 查找您所在的景点到其他景点的最短路径
3. 查找任意两个景点之间的最短路径
4. 修改景点信息
5. 增加景点信息
6. 删除景点信息
7. 增加道路
8. 删除道路
9. 退出管理系统

管理员系统

HBU景点列表:

- 1 : 操场
- 2 : 图书馆
- 3 : 体检中心
- 4 : 校门北口
- 5 : 银杏景观
- 6 : 邯郸音乐厅
- 7 : 餐厅
- 8 : 花园景观
- 9 : 校门东口
- 10 : 信息学部
- 11 : 网计学院
- 12 : 校门南口

你要看那个说号:

景点描述

- 1 : 操场
- 2 : 图书馆
- 3 : 体检中心
- 4 : 校门北口
- 5 : 银杏景观
- 6 : 邯郸音乐厅
- 7 : 餐厅
- 8 : 花园景观
- 9 : 校门东口
- 10 : 信息学部
- 11 : 网计学院
- 12 : 校门南口

请输入您所在的景点编号:1

- 280米 : 图书馆<-操场
 350米 : 体检中心<-操场
 200米 : 校门北口<-操场
 480米 : 邯郸音乐厅<-操场

最短路径查询

```
2 : 图书馆
3 : 体检中心
4 : 校门北口
5 : 银杏景观
6 : 邯郸音乐厅
7 : 餐厅
8 : 花园景观
9 : 校门东口
10 : 信息学部
11 : 网计学院
12 : 校门南口
您要删除哪个景点?
1
删除的景点为: 操场真的要删吗? 输入1 for sure:
1
删除景点ING
已经删完了
, 1s后回到主界面_
```

删除功能

韩旭051