

⇧ 细化需求

⇧ 整体目录

细化需求

- 1.整体设计
- 2.各个模块以及涉及到的函数
 - 2.1主模块
 - 2.2功能模块
 - 2.2.1创建图的邻接矩阵的存储结构
 - 2.2.2浏览图中任一景点介绍
 - 2.2.3输出所有景点
 - 2.2.4游客系统
 - 2.2.5管理员系统
 - 2.2.6查找游客所在景点与其他景点的距离
 - 2.2.7查找游客指定的两个景点间的最短路径长度
 - 2.2.8修改景点信息
 - 2.2.9添加景点
 - 2.2.10删除景点
 - 2.2.11 添加道路
 - 2.2.12 删除道路
 - 2.2.13输出路线信息

⇧ 1.整体设计

为了结构清晰，将模块分为**几个头文件**来存放。

`<first.h>` 头文件定义一些结构体和常量

`fun.h` 来存放一些功能函数。

2. 各个模块以及涉及到的函数

2.1 主模块

```
1 | int main()
```

主要用于判断用户权限信息，以及主要的交互面板。

采用命令行的格式进行输出。

2.2 功能模块

2.2.1 创建图的邻接矩阵的存储结构

```
1 | void create(void)
```

采用文件的形式存入顶点和边的信息。

数据结构：图的邻接矩阵。

2.2.2 浏览图中任一景点介绍

```
1 | void introduct(void);
```

正常交互逻辑

2.2.3 输出所有景点

```
1 | void showInfo(void);
```

算法：循环输出

2.2.4 游客系统

```
1 | void visiter(void);
```

条件变量进行判断是游客还是管理员



2.2.5 管理员系统

```
1 | void Administrator(void);
```

条件变量进行判断是游客还是管理员



2.2.6 查找游客所在景点与其他景点的距离

```
1 | void DijkstraHeap(void);
```

算法：Dijkstra算法。目前初步定的优化方案是**采用最小堆**进行优化



2.2.7 查找游客指定的两个景点间的最短路径长度

```
1 | void Floyd(void);
```

算法：采用图的存储结构，搭配最短路径算法。



2.2.8 修改景点信息

```
1 | void modifyInfo(void);
```

指定景点进行修改信息，修改值。



2.2.9 添加景点

```
1 | void addInfo(void);
```

在景点数组上进行扩容，然后增加景点的信息。



2.2.10 删除景点

```
1 | void delInfo(void);
```

通过输入景点的值，查找数组，然后删除指定数组的值。

2.2.11 添加道路

```
1 | void addPath(void);
```

通过输入道路的值，然后循环处理数组里面的结构，最后添加道路成功。

2.2.12 删除道路

```
1 | void delPath(void);
```

通过输入要删除道路的值，循环处理数组里面的结构，最后删除成功。

2.2.13 输出路线信息

```
1 | void printPath(int[], int, int);
```

算法: **使用递归的方法**输出路线!

以上是本次校园导航系统的大致函数设计!