

## 一、需求分析

MainPage: 用于基本的可视化展示, 可以根据用户需求进行跳转。

Page\_1: 实现对景点基本信息的查询。

Page\_2: 实现对景点间路径的查询, 包括最短路径和全部路径。

Page\_3: 时间对路径、景点的增删改。

Returnpage: 实现在每次调用函数后决定是否退出本系统或者继续查询。

## 二、概要设计

在 ReadCostAndPlace 包含在 guild.h 头文件中, 用于读取 cost.txt 和 place.txt。初始化都路径权值表、名称表、备注表等。

当然我们可以根据模块 Page3 进行进一步增删改。

权值表使用 `vector<vector<int>> cost(MaxV, vector<int>(MaxV, Maxint))`;

其他使用 string。

综合使用了例如 stack、queue 等多种数据结构。

## 三、详细设计

代码附件。

最短路径使用 Dijkstra 算法。全部路径使用 DFS 进行搜索。

在 VE 增删改中需要注意是否 over, 或者已经存在相应边点。

## 四、调试分析

### 1. 查询展示

```
welcome to Fudan University!
Here is the Guild system for you.
Please wait for seconds.
Verison 1.00 author: Zhenghao Ji

Please enter a number
1:查询景点相关信息.
2:查询景点间路径.
3:修改景点信息.
q:关闭本系统.

1
0:GHL
1:LDS
2:LIBRARY1
3:LIBRARY2
1
LDS : BELONGTOSOM
r:返回主页面
q:退出本系统
任意键继续查询.
r
Please enter a number
1:查询景点相关信息.
2:查询景点间路径.
3:修改景点信息.
q:关闭本系统.
```

返回主页面进行路径功能展示:

0-3 最短路径:

```

2
Page2.
Please enter the start and end place.
0 3
Please enter 1 for the shortest path and 0 for all paths.
1
GHL->LIBRARY1->LIBRARY2
All sum distance: 2
r:返回主页面
q:退出本系统
任意键继续查询.

```

所有路径：共四条

```

Page2.
Please enter the start and end place.
0 3
Please enter 1 for the shortest path and 0 for all paths.
0
0->1->2->3 Sum is 4
0->1->3 Sum is 6
0->2->1->3 Sum is 6
0->2->3 Sum is 2
r:返回主页面
q:退出本系统
任意键继续查询.

```

展示中最短路径使用相应名称展示，所有路径使用数字展示。做两种展示表示均可。

增加景点 AAA 和备注 BBB

```

Page3.
Please enter a number
1:增加路径.
2:删除路径.
3:更新路径.
4:增加景点.
5:删除景点.

4
Please enter the name and the note.
AAA BBB
r:返回主页面
q:退出本系统
任意键继续查询.
r

```

```

Please enter a number
1:查询景点相关信息.
2:查询景点间路径.
3:修改景点信息.
q:关闭本系统.

1
0:GHL
1:LDS
2:LIBRARY1
3:LIBRARY2
4:AAA
4
AAA : BBB
r:返回主页面
q:退出本系统
任意键继续查询.

```

再次查询发现已经添加。

```

Please enter a number
1:增加路径.
2:删除路径.
3:更新路径.
4:增加景点.
5:删除景点.

1
Please enter the start,end and the cost.
0 3 9
The path has been added.
r:返回主页面
q:退出本系统
任意键继续查询.

```

```

Please enter the start and end place.
0 3
Please enter 1 for the shortest path and 0 for all paths.
0
0->1->2->3 Sum is 4
0->1->3 Sum is 6
0->2->1->3 Sum is 6
0->2->3 Sum is 2
0->3 Sum is 9
r:返回主页面
q:退出本系统
任意键继续查询.

```

增加路径 0 3 9

再次查询 0-3 全部路径变为五条。

```

2
Please enter the start,end.
0 2
The path has been deled.
r:返回主页面
q:退出本系统

```

删除路径 0-2

```

Please enter 1 for the shortest path and 0 for all paths.
1
GHL->LDS->LIBRARY1->LIBRARY2
All sum distance: 4
r:返回主页面
q:退出本系统
任意键继续查询.

```

最短路径改变 原本长度为 2 的路径消失。

```

Page3.
Please enter a number
1:增加路径.
2:删除路径.
3:更新路径.
4:增加景点.
5:删除景点.

5
Please enter the num.
3
r:返回主页面
q:退出本系统
任意键继续查询.

```

```

1
0:GHL
1:LDS
2:LIBRARY1
3:AAA

```

删除景点 3 可再次查询。

## 五、课程设计总计

由于使用了邻接矩阵且在删除某景点时候，需要将其他路径权值进行移动，将其原本数据进行覆盖导致时间复杂度较高为  $O(n^2)$

增加景点则是  $O(1)$ 。

对于边的增删改同样均为  $O(1)$ 。

最短路径使用 Dijkstra 时间复杂度  $O(n^2)$ ，如果是不经常修改，可以使用 Floyd 算法进行全部计算。尽管在时间复杂度上并未有优势，但是在现实生活中使用更方便不需要每次都计算而只需要读取数组即可。

数据结构设计不够完善：Place 和 Note 使用了两个 `vector<string>` 没有将其有效的进行合并成一个 node。且使用临界矩阵虽然简单但是时空复杂度较高。后期有这个意识，鉴于代码完成度较高遂放弃进行更多优化。

## 六、参考资料

无其他人代码，均为独立完成。