**数据结构与算法程序设计**

**地铁线路查询系统**

# 1 项目要求

|  |  |  |  |
| --- | --- | --- | --- |
| 项目名称 | 地铁线路查询系统 | 项目类型 | 应用软件类 |
| 项目难度 | 中等 | 素材资源 | 无（../res） |
| 使用工具 | 不限 | 编译系统 | Windows、Linux |
| 硬件需求 | 无 | 程序语言 | 不限 |
| 知识点 | 结构体/类、链表、队列、查找、排序、递归、函数、文件操作、键盘操作等 | | |
| 项目描述 | 本项目是对地铁线路信息的简单模拟查询，以完成建立地铁路线信息、修改地铁路线信息、删除地铁路线信息和查询地铁线路信息等功能。 | | |
| 功能实现 | * 程序主函数：系统入口，监控程序运行。 * 创建地铁线路功能：创建多个地铁线路比如1号线、2号线等 * 各个地铁口信息查询：查询地铁口信息 * 两个地铁口最短路径查询：两个地铁口之间最短距离 * 两个地铁口所有路径查询：两个地铁口所有路径 * 系统管理员及后台管理 * 退出查询系统：退出系统 | | |
| 提交材料 | 实训报告、可运行的程序 | | |

# 2 技术方案建议

### 2.1数据结构的选择

按具体需求自选数据结构

### 2.2算法参考

1．地铁网的实现是通过图来实现的；

2．两个地铁口最短路径模块是用floyed算法通过一个图的权值矩阵求出它的每两点间的最短路径矩阵；

3．求地铁网的关节点模块是通过深搜优先生成树来判定。从任一点出发深度优先遍历得到优先生成树，对于树中任一顶点Ｖ而言，其孩子节点为邻接点。

4．两个地铁口所有路径的查询依然是采用的DFS算法，每一次将起点压入栈内，并将其标记为已访问，然后检查其是否有未访问的邻接点，然后将其邻接点作为起点采取深度搜索做相同的操作，若遇到路不通的时候，依次将栈内的点推出，找其下一个未访问的邻接点，直到找到终点，算法结束。

# 3 审查标准

### 3.1 审查要点及评审标准

|  |  |  |  |
| --- | --- | --- | --- |
| **项目审查要点** | | **分值** | **得分** |
| **基本** | 界面显示、菜单交互 | 10 |  |
| 创建地铁线路功能 | 20 |  |
| 各个地铁口信息查询 | 10 |  |
| 两个地铁口最短路径查询 | 15 |  |
| 两个地铁口所有路径查询 | 15 |  |
|  系统管理员及后台管理 | 10 |  |
| 退出查询系统：退出系统 | 10 |  |
| **扩展** | 删除地铁网 | 10 |  |
| **项目总分** |  | 100 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **评分项** | **评分标准** | **分值** | **得分** |
| **项目得分** | 具体见项目审查要点，若不能运行或编译则为0分，分值为项目总分\*40% | 0-40 |  |
| **文档得分** | 观点明确，论据充足，格式规范，结构清晰完整，能较好的体现项目内容 | 8-10 |  |
| 内容充实，格式规范，结构清晰，在表达或格式上略有欠缺 | 5-8 |
| 内容完整，格式有欠缺或结构不清晰 | 3-5 |
| 内容不完整，结构不清晰，未按照文档规范书写 | 0-3 |
| **用户体验得分** | 实现项目效果并有创新，界面美观，用户体验良好 | 8-10 |  |
| 实现项目基础效果，界面美观度一般，用户体验一般 | 5-8 |
| 勉强实现项目效果，用户体验较难 | 3-5 |
| 未实现项目效果，用户无法体验 | 0-3 |
| **答辩** | 思路清晰、有理有据、对于重点解释清晰，有深度和创新 | 30-40 |  |
| 思路清晰，能正确回答问题，但缺少深度 | 20-30 |
| 问题需提示才能答出或回答浅显 | 10-20 |
| 不能正确回答问题 | 0-10 |
| **总分** | 优（90以上）良（80-90）中（70-80）及格（60-70）不及格（60以下） | |  |