

地铁换乘

请用C++和面向对象程序设计的方法完成一个提供地铁换乘方案的任务。程序能够从给定的csv文件中读取各条地铁线路的站点和换乘信息，根据用户输入的起点和终点（包含地铁线路和站点），给出乘车方案。

csv 文件给定的线路信息中，站点 ID 升序排列，同一线路的站点按照前后相邻关系顺序排列。

搜索乘车路线时，从某一个站点出发，可以向前后相邻站点展开搜索；如果该站点有换乘站，还可以跳转到另一条线路继续搜索。为避免重复搜索，需标记某站点是否已经访问过。建议使用递归完成搜索功能。

程序的逻辑流程

根据任务说明，先设计程序的整体逻辑流程。这一步很关键，本应由同学们自行完成。但是，为了规范大作业的实现，将本学期所学内容尽可能多地用于大作业，给出一个程序框架如下：

```
int main()
{
    ifstream infile("metro.csv");
    path p(infile);           //从文件中读取线路图

    cout << "Enter starting station,Line *-ending station,Line *:" << endl;
    p.read_start_end(cin);    //从用户输入读取起点和终点信息
    if (!p.isvalid()) {
        cout << "Start/End station not included in the metro map";
        return 1;
    }

    if (p.findpath()) {       //找到乘车路线
        cout << "====Route found====" << endl;
        p.optimize();        //优化线路,去掉连续换乘
        p.output();          //输出乘车路线
    }
    else
        cout << "====No route found====" << endl;

    return 0;
}
```

设计类和类的成员

根据程序的逻辑流程，设计所需要的类和类的接口：

1. 站点类 Station

数据成员：站点名称, 所属线路, 换乘站点及线路（可能有多条），访问标志

成员函数：

- 1) 构造函数（可能有多条）
- 2) 访问器函数（可能有多条）
- 3) 从用户输入中读取站点信息
- 4) 设置站点的访问状态

与站点类密切相关的全局函数：判断两个站点是否为同一站点（重载==运算符）

2. 路径类 Path

数据成员：已知线路图，起点信息，终点信息，起点-终点的换乘路线

类型成员：线路图或乘车路线所包含的站点数量所对应的类型

成员函数：

- 1) 构造函数（可能有多个）
- 2) 从用户输入初始化起点和终点
- 3) 判断起点和终点是否包含在线路图中
- 4) 搜索乘车路线
将递归搜索封装成私有成员函数，在公有成员函数中调用私有成员函数。
- 5) 优化乘车路线
优化线路，去掉连续换乘（参见附件）
- 6) 输出乘车路线

可根据需要，在以上两个类中自行添加数据成员、类型成员和函数成员，或者定义更多的类。

附件

1. metro.csv：已知地铁线路的站点和换乘信息。注意：请勿用Excel打开，它会将类似11/17的换乘数据转换成日期格式11月17日。
2. example.png：运行样例
3. original.png, optimized.png：线路优化前和优化后的运行结果