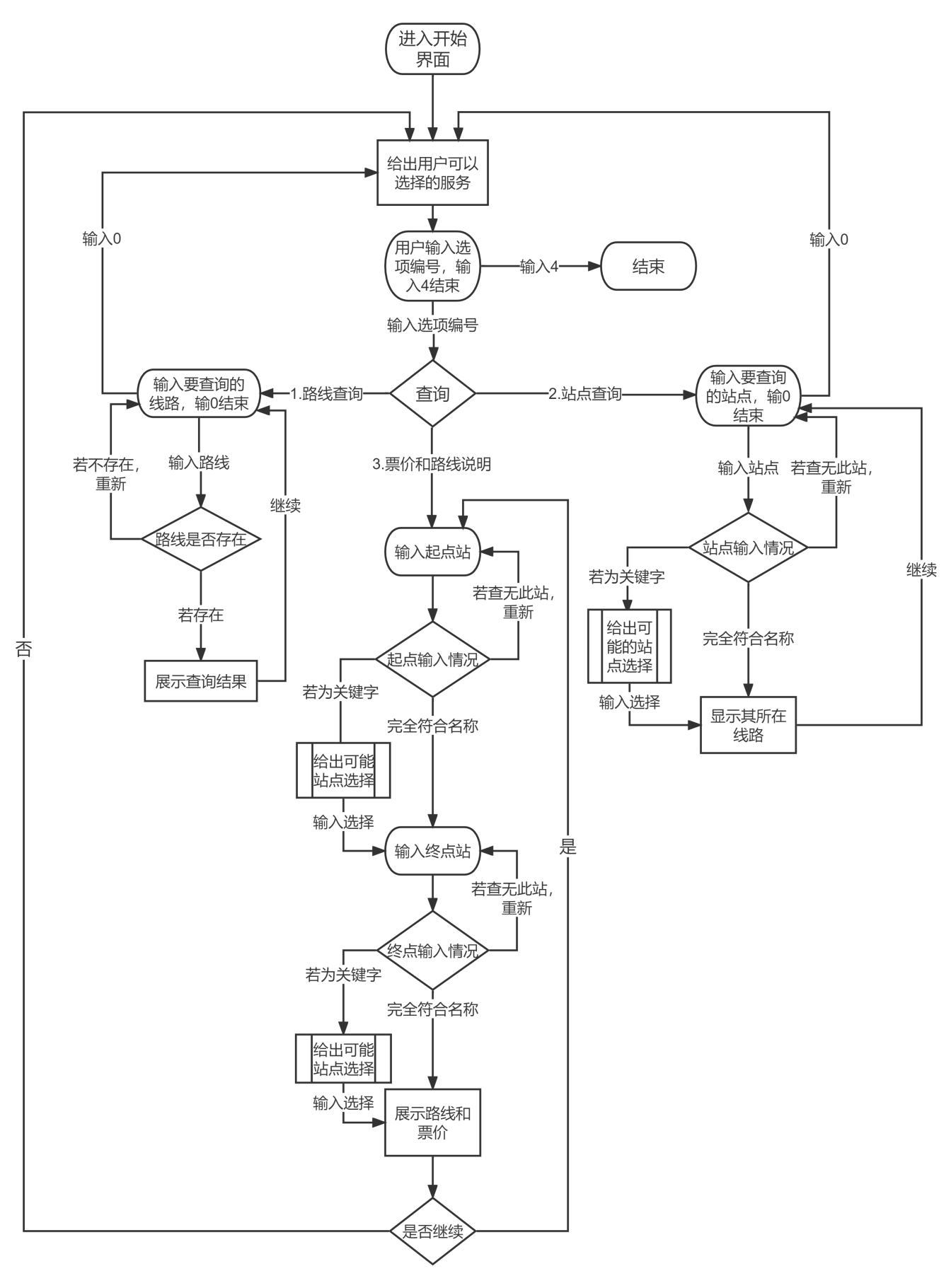
**项目设计**

**1.实验内容概述**

本实验实现简易的南京地铁票务系统。在控制台终端界面进行操作，需要与用户良好交互，实现功能包括：线路查询（显示路线上所有站点），站点查询（支持关键字搜索和子串匹配，供用户选择），输入起点和终点反馈路线规划（在距离相差不太大情况下，给出多种可能路线）给出说明和票价（按照最短路线计算）。

1. **程序框架设计**
2. 简易的欢迎界面和用户指引选项。提供选项有：线路查询、站点查询、购票和路线规划、退出票务系统。
3. 线路查询：首先显示目前运营线路，根据用户输入，展示相应路线或者告知输入错误。可进行多次查询，给予退出线路查询模块的提示——输入0。
4. 站点查询：提示用户应该输入站点，当用户输入站点不存在时，提示重新输入，直至正确或者出现关键字；有关键字时通过关键字索引，给用户列出选择来找到想要查询的；正确或多一个“站”字都可以输出其所在线路。
5. 票价查询和输出规划路线：提示用户输入起点站和终点站，并且每次输入都要判断是否存在，或者判断是否为关键字并给出选择。允许起始终点相同。规划路线不止一条最短的，还有其他接近的距离的方案给用户选择，或者考虑换乘最少等其他方面的。票价计算用区间计算，取最短的计算价格。具体实现后面说明。

**框架设计图如下：**

****

1. **主体模块函数说明**
2. 文件预处理：为了读取文件操作方便，本人将文件分解成一条线路一个.txt文件，格式是一行一个“站点名 到下一个站点距离”，终点站的到下一站点距离定为0。
3. 设立两个类：mylines（置于lines.h头文件）和mymap（置于Metro\_Map.h头文件里）。

mylines包括成员：struct station{string name; int cond=0;int distance=0;int sub\_distance = 0;};分别代表：站点名称、记录查询状态、到下一站距离、到上一站距离。string类型的number（几号线），int型的stat\_num（该线路上多少站）。

成员函数有station\*类型的creatline(string filename)函数，用于线路的初始化，返回值指向station类型的数组，形参代表文件名称。路线创建用文件读取函数ifstream in\_file(filename, ios::in)，in\_file重载读取。

mymap类包含mylines的线路数组L[]。成员函数包括init函数（初始化各线路），无形参；search\_line用于线路查询，展示路线选项编号，直接按照用户搜索的编号匹配，无形参。

mymap类里的search\_station函数用于站点查询，无形参，内部调用output\_station(string s)输出函数，带有交互，关键词搜索用的库函数查找string2子串位置string1.find(string2)，在所有站点中匹配子串，输出给用户选择。线路查找直接在第一次查找到后进行继续查找其他线路上对应站点，保存对应线路，达到输出换乘情况效果。为了记录，每个选项信息将存在结构体数组中，包括站点名称和换乘线路。

1. 线路规划：接收起点和终点。在mylines类里面，线路存法用“链式前向星”方法。初始化构造stat\_edge的结构体，包括int类型的to,val,next;分别代表下一站点标号，线路长度（权），该起点的下一条边的编号。存图效果为有向边组成的图。考虑到实际情况，需要两个方向各存一遍，因此可看作是无向图。用int型head数组存从第i点出发的第一条边的位置。

路线查找用dfs（深度优先算法）从给定起点遍历到终点，若能到终点，则记录找到的路径及距离，每次比较取最小即产生最短路径值。环形路线则计次，不让循环多次。

1. 票价计算：参照老师的票价计算的文档，有待商榷的是第九档的临界计算。本人算法是在70公里票价基础上，增量未满14公里增加1元，再进行每14公里加1元，比如乘坐75公里的票价10元，84公里的票价11元，88公里的11元，以此类推。