实习报告

**题目**：编制一个交通咨询模拟程序，为旅客提供不同要求下的路线规划

**组员**：吕星宇、胡康、赵昱杰 **成稿日期**：2022.06.25

一、需求分析

1. 问题描述

处于不同目的的乘客对交通工具有不同的要求，比如在旅途中的时间尽可能短，旅费尽可能的省，中转次数最少等。编制一个全国城市间的交通咨询程序，为旅客提供两种或三种最优决策的交通咨询。

2. 功能描述

* **显示列车时刻表**

用户选择功能“显示列车时刻表”，输出现有的列车时刻表，支持翻页的功能。

* **显示飞机航班表**

用户选择功能“显示飞机航班表”，输出现有的飞机航班表，支持翻页的功能。

* **编辑列车时刻表**

用户选择功能“编辑列车时刻表”，可以在列车时刻表中添加或删除车次。

* **编辑飞机航班表**

用户选择功能“编辑飞机航班表”，可以在飞机航班表中添加或删除航班。

* **进行路线规划**

用户输入“起点”“终点”“交通方式”，输出三种决策方式的推荐路线。

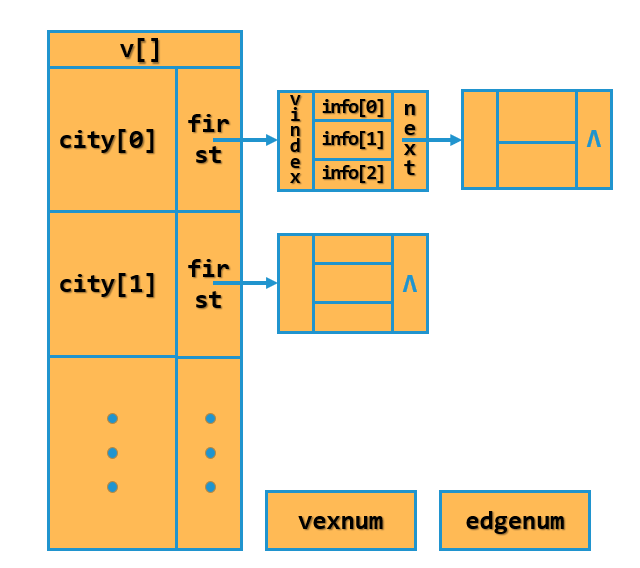
3. 测试数据

见文件TrainTable.txt，FlightTable.txt。

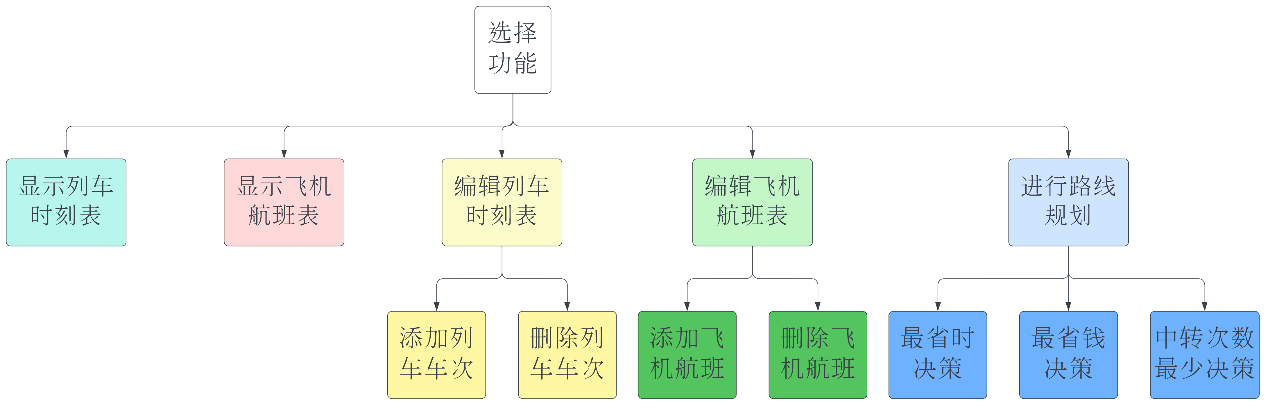
二、设计框架

1. 数据结构

使用邻接表，在表结点中新加一Info域存储一趟车次/航班的相关信息，包括tag标签（标记这是一趟车次还是航班）、编号、价格、出发时间、到达时间、旅程用时。邻接表示意图如下图所示：

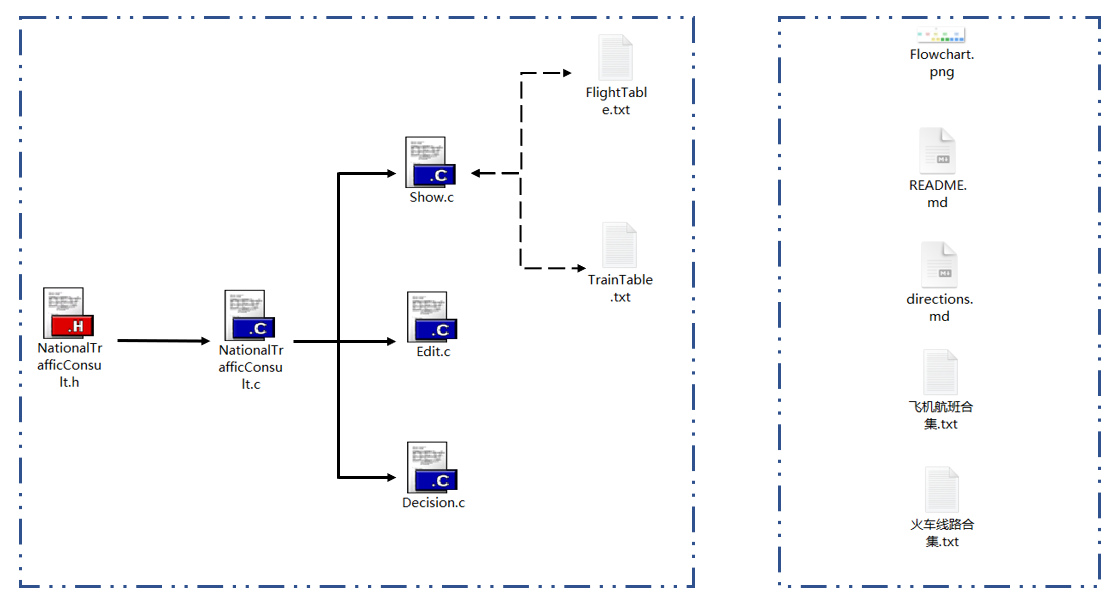


2. 执行流程



3. 相关文件

文件结构如下图所示：



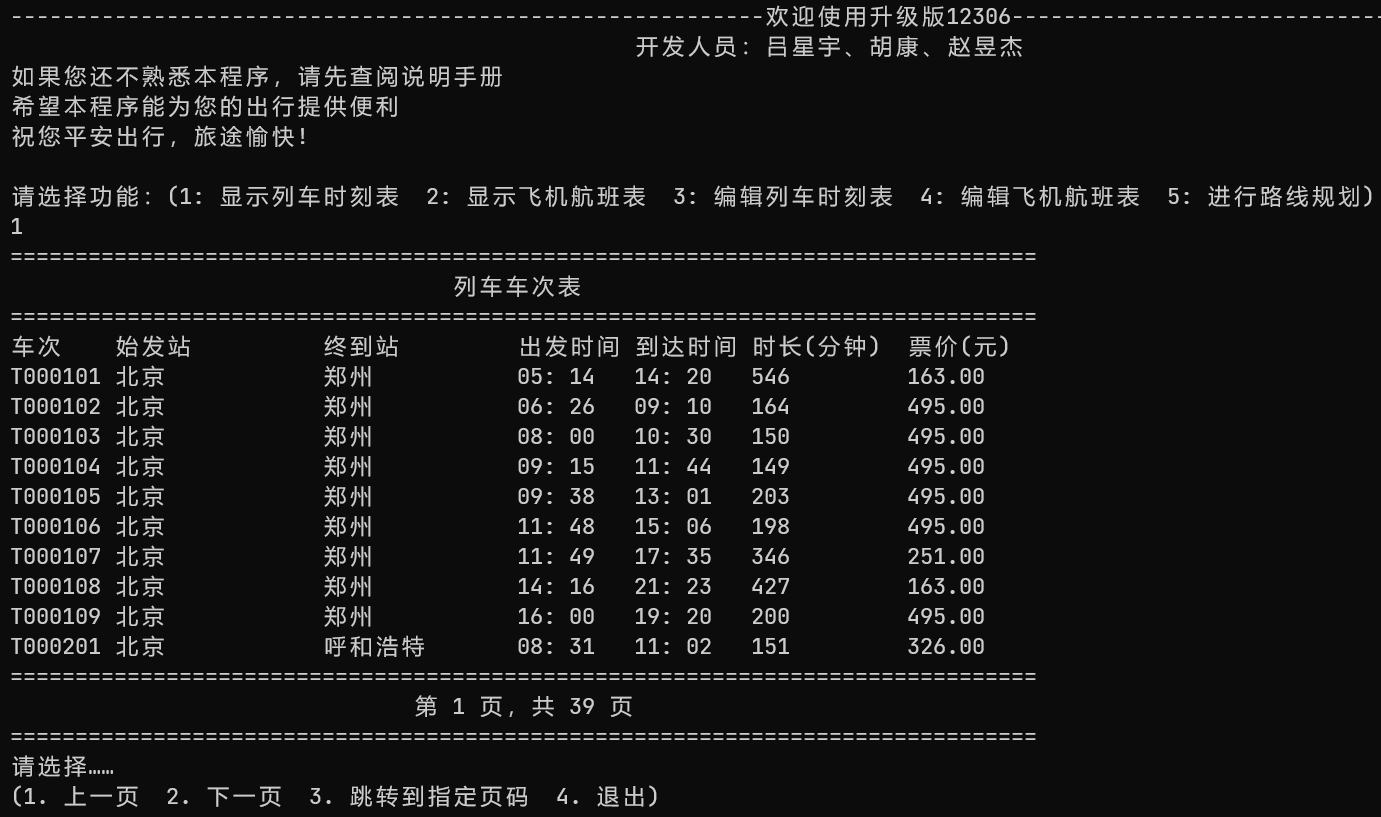
其中，头文件NationalTrafficConsult.h保存了数据结构和主要函数的定义，可以查看该文件大致了解本程序的执行流程。NationalTrafficConsult.c中含有主函数，根据用户输入，通过switch语句选择调用的函数。Show.c文件用于和FlightTable.txt及TrainTable.txt交互，得到当前所有线路信息。Edit.c文件用于向交通图中添加和删除线路，同时也保存到FlightTable.txt及TrainTable.txt中，这样重启程序时上次操作结果仍然保留。Decision.c文件用于进行路线决策，向用户给出三种不同决策方式下的最优路线推荐。

其它文件包括：Flowchart.png，展示程序的执行流程；README.md，对程序整体架构和代码规范做说明；directions.md，用户手册；飞机线路合集.txt和火车线路合集.txt，含有可批量添加的线路。

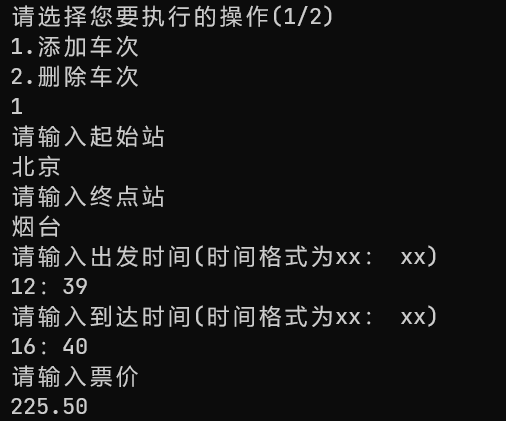
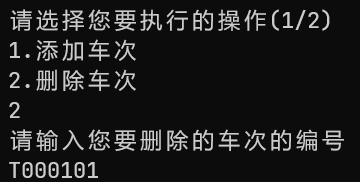
三、测试运行结果

① 打开程序即导入已保存线路；

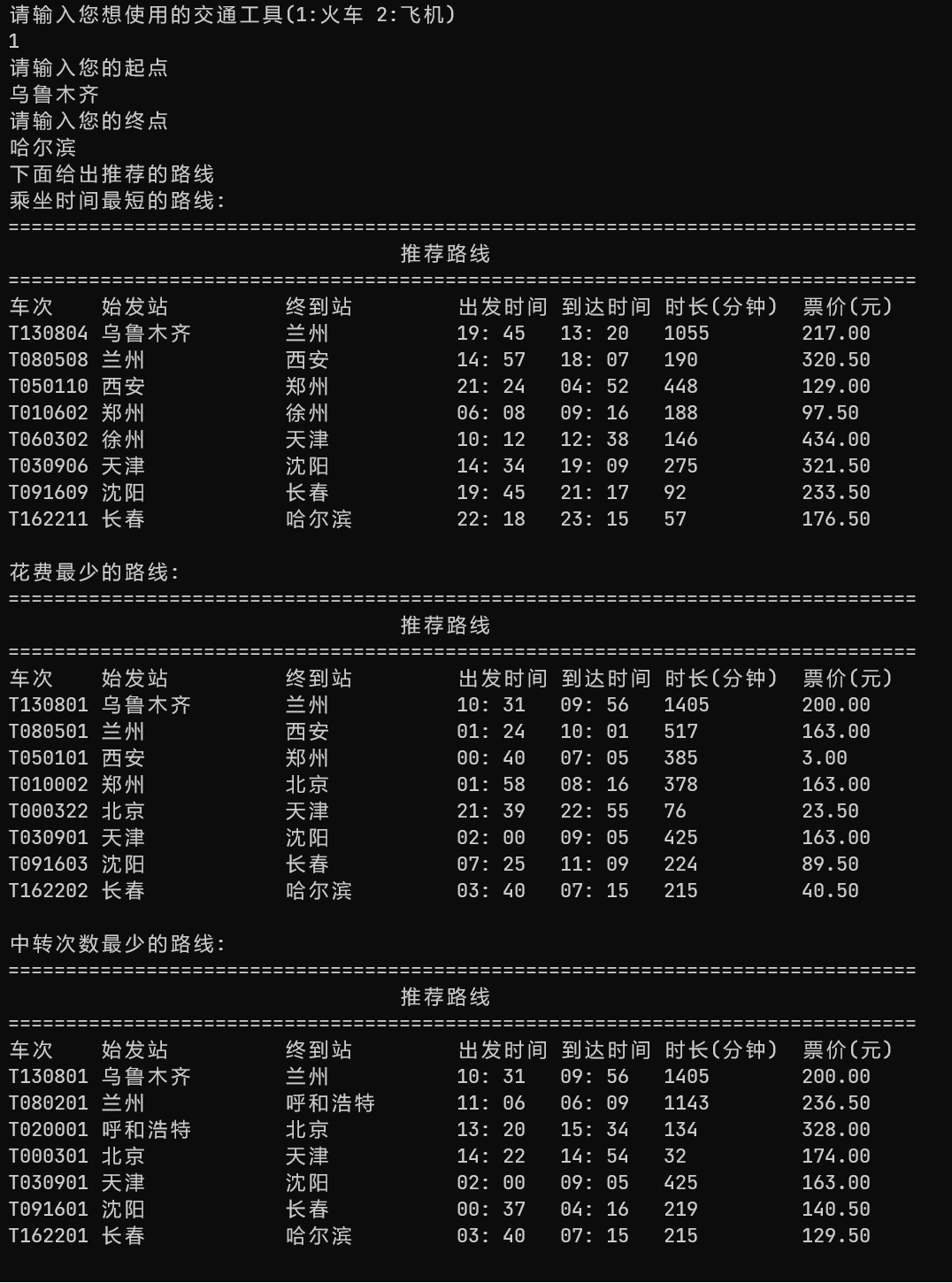
② 功能1：显示列车时刻表/飞机航班表（以列车时刻表为例）；



③ 功能2：编辑列车时刻表/飞机航班表（以向列车时刻表中新增、删除车次为例）；

④ 功能3：进行路线决策。

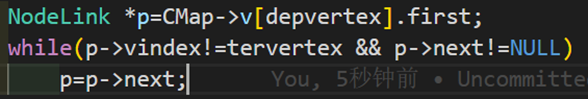


四、调试分析

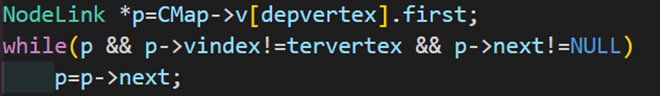
1. 问题与解决方式

① **空指针segmentation fault问题**

在增加列车时，针对起始站、终点站均已存在且两地原本没有列车的情况下，实际操作就是在起始站所在的链表的末尾增加一个终点站的节点。为了让一个中间变量指针指向起始站所在链表末尾，将循环设置为：



但在运行过程中出现了segmentation fault。分析后发现，若起始站是尚未与任何城市之间有列车，那么它的表头指针指向NULL，故p指向NULL，那么p->vindex便会报错。为了解决该问题，只需先判断p是否指向NULL：



这样便避免了该问题。

② **最少时间决策权值变化问题**

在解决最短时间花费的决策问题中，存在时刻之间的交错，比如北京到郑州的到达时间是15：00，要乘坐郑州到乌鲁木齐的车的出发时间是14：00，则需要等到第二天才能出发，这样会导致等待时间的增加。

经过讨论，发现问题的根源在于起点的出发时间不确定，所以对这部分算法进行了调整，原先是将起点作为源点，现在将从起点发出的车的每一个线路的终点作为源点，将到达时间作为此源点的出发时刻（这样是固定的），对每个源点均使用Dijkstra算法，再选择其中用时最少的。

2. 算法的性能分析

① **时间复杂度**：核心算法利用Dijkstra算法，为。

② **空间复杂度**：主要存储结构为邻接表，所占空间为；此外还有几个数组辅助结构，所占空间为常数级。

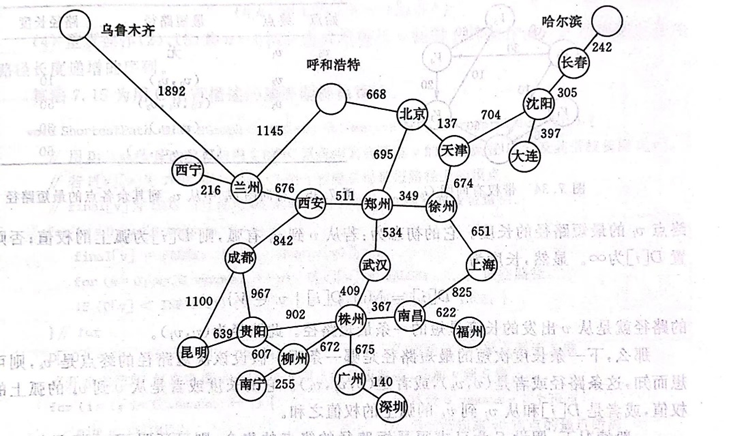
五、用户使用说明

程序开始运行时，会有1、2、3、4、5五种选项对应五种功能，用户可以根据自己的需求选择不同的功能。

本程序根据全国交通网络，在30座城市中设立了800余条火车线路和飞机航班，每一条火车线路与飞机航班都有专属编号，为“XAABBCC”，编号规则为：

* X为T或F，T表示火车，F表示飞机；
* AA表示始发站城市编号；
* BB表示终点站城市编号；
* CC表示相同城市间列车/航班序号。

初始情况下，30座城市的交通连接情况为：



用户可以选择选项1来查看初始的火车线路，也可以选择选项2来查看初始的飞机航班。由于列车/航班时刻表过大，用户可通过翻页功能对其进行查看。

在初始30座城市和800余条线路的基础上，用户可以按照个人意愿增加城市/线路。用户可以选择选项3来修改火车线路，也可以选择选项4来修改飞机航班。

无论是修改火车线路还是飞机航班，都可以选择增加或删除两种功能：

* 若要**增加线路**，用户需要按照提示依次输入始发站、终点站、起始时间、终止时间、票价。程序会自动生成线路编号并保存至线路表；
* 若要**删除线路**，用户只需输入编号，系统便会完成删除。

本程序的核心功能是进行路线规划。用户可以选择选项5来规划最便利、最实惠的路线。根据出行工具，用户可以选择火车或飞机，然后输入始发地和目的地。程序会规划出乘坐时间最短、花费最少、中转次数最少三种最优方案。

六、附录

您可在<https://github.com/Zemdalk/DS-project-work>的子文件夹“NationalTrafficConsult”下查看全部代码和相关文件。