

# Data structures Final Assignment

Yaoming Shi, Hanlin Qian, Yizhou Xue

UCAS

2025.6.25

## 5.8⑤ 全国交通咨询模拟

### 【问题描述】

出于不同目的的旅客对交通工具有不同的要求。例如,因公出差的旅客希望在旅途中的时间尽可能短,出门旅游的游客则期望旅费尽可能省,而老年旅客则要求中转次数最少。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。

### 【基本要求】

- (1) 提供对城市信息进行编辑(如:添加或删除)的功能。
- (2) 城市之间有两种交通工具:火车和飞机。提供对列车时刻表和飞机航班进行编辑(增设或删除)的功能。
- (3) 提供两种最优决策:最快到达或最省钱到达。全程只考虑一种交通工具。
- (4) 旅途中耗费的总时间应该包括中转站的等候时间。
- (5) 咨询以用户和计算机的对话方式进行。由用户输入起始站、终点站、最优决策原则和交通工具,输出信息:最快需要多长时间才能到达或者最少需要多少旅费才能到达,并详细说明依次于何时乘坐哪一趟列车或哪一次班机到何地。

### 【测试数据】

参考教科书 7.6 节图 7.33 的全国交通图,自行设计列车时刻表和飞机航班。

【实验报告二】

# 总体设计与功能

- 支持火车/飞机数据的批量导入。
- 火车建模：分层图最短路，支持不同优先级。
- 飞机建模：直接对起点和终点之间连边，跑最短路。

```
data > TrainInfo.tbl > data
1 2602|1|秦皇岛|北戴河|0|17:20|17:38|1|18|0|7.00|NUL
2 2602|2|北戴河|滦县|3|17:41|18:45|1|64|0|13.50|NUL
3 2602|3|滦县|唐山|17|19:02|19:57|1|55|0|21.50|NUL
4 2602|4|唐山|芦台|9|20:06|20:34|0|28|0|NULL|NULL|
5 2602|5|芦台|天津|3|20:37|21:52|1|75|0|38.50|NULL
6 2602|6|天津|北京|17|22:09|23:48|1|99|0|54.50|NUL
7 2602|7|北京|张家口南|30|00:18|03:43|1|205|0|75.00
8 2602|8|张家口南|阳高|6|03:49|05:36|1|107|0|86.00|
9 2602|9|阳高|大同|2|05:38|06:31|1|53|0|93.00|NULL
10 4060|1|塔河|翠岗|0|07:56|08:25|0|29|0|NULL|NULL|
11 4060|2|翠岗|碧州|1|08:26|08:38|0|12|0|NULL|NULL|
12 4060|3|碧州|新林|1|08:39|09:07|1|28|0|6.00|NULL|
13 4060|4|新林|林海|2|09:09|09:22|1|13|0|7.50|NULL|
14 4060|5|林海|新天|2|09:24|10:02|0|38|0|NULL|NULL|
15 4060|6|新天|劲松|2|10:04|10:19|0|15|0|NULL|NULL|
16 4060|7|劲松|古源|2|10:21|10:37|0|16|0|NULL|NULL|
17 4060|8|古源|小扬气|1|10:38|10:51|1|13|0|16.50|NUL
18 4060|9|小扬气|加格达奇|2|10:53|11:40|1|47|0|20.50|
19 6074|1|平凉|平凉南|0|13:10|13:25|1|15|14|1.00|NUL
20 6074|2|平凉南|庙庄|2|13:27|13:36|1|9|7|2.00|NULL|
21 6074|3|庙庄|崇信|2|13:38|14:06|1|28|16|2.00|NULL
22 6074|4|崇信|安口窑|2|14:08|14:23|1|15|12|3.00|NUL
23 6074|5|安口窑|神峪河|3|14:26|15:17|1|51|14|4.00|N
24 6074|6|神峪河|火烧寨|20|15:37|15:53|1|16|14|4.00|
25 6074|7|火烧寨|史家铺|6|15:59|16:16|1|17|14|6.50|N
26 6074|8|史家铺|陇县|2|16:18|16:34|1|16|12|6.50|NUL
27 6074|9|陇县|娘娘庙|2|16:36|16:49|1|13|11|7.50|NUL
```

# 火车部分建模与查询

- **分层图建模**：每个车站每个车次拆点，分两层，支持换乘建边。
- **边权设计**：一类边为区间时间/票价，二类边为换乘间隔。
- **最短路算法**：Dijkstra，支持时间优先/价格优先。
- **换乘规则**：可设定是否允许换乘及最小换乘间隔。
- **查询接口**：支持城市/车站为起终点，输出详细方案。

# 飞机部分建模与查询

- **点对点建图**：每个城市为节点，航班为有向边。
- **支持一次中转**：简单 bfs，允许中转查询。
- **多关键字查询**：可选时间优先或价格优先。
- **批量导入**：自动识别城市名并建表。
- **详细输出**：每段航班号、起降城市、时间、价格。

# 数据结构与核心实现

- **哈希表**：城市名、站名映射唯一 ID，支持中文。
- **链式前向星**：存储火车分层图、飞机航班图。
- **小根堆**：Dijkstra 最短路，支持多关键字比较。
- **全局变量**：统一管理所有城市、站点、图结构。
- **失效标记**：通过 expire 字段实现逻辑删除/恢复。对每个 Station 和 Train\_id 维护一个 expire 字段，表示该车次是否失效。

# 用户交互界面

```
==== National Transport System ====  
1. Add train station  
2. Delete train station  
3. Add train  
4. Delete train  
5. Add flight  
6. Delete flight  
7. Query train (city to city)  
8. Query train (station to station)  
9. Query flight (city to city)  
0. Exit  
=====
```

控制台交互界面

# 运行结果——火车车次加入与查询

```
g++ -c src/hash.cpp -o build/hash.o -Wall -Wextra -O2 -I include
Linking with: build/app.o build/global.o build/graph.o build/heap.o build/main.o build/modify.o build/plane.o
g++ build/app.o build/global.o build/graph.o build/heap.o build/main.o build/modify.o build/plane.o
./Project

===== National Transport System =====
1. Add train station
2. Delete train station
3. Add train
4. Delete train
5. Add flight
6. Delete flight
7. Query train (city to city)
8. Query train (station to station)
9. Query flight (city to city)
0. Exit

Enter your choice: 1Enter station name: 北京
Enter city name: 北京
Station added successfully.
Input: 1
Enter station name: 葫芦岛
Enter city name: 葫芦岛
Station added successfully.
Input: 3
Enter train ID: G123
Enter number of stations: 2
Enter 2 station names (separated by space):
北京 葫芦岛
Enter 1 segments (depart_time(HH:MM) arrive_time(HH:MM) price):
08:00 10:30 120.00
Train added successfully.
Input: 7
Enter start city: 北京
Enter end city: 葫芦岛
Time first(0) or price first(1): 0
Allow transfer? (0/1): 0
Minimal transfer interval (minutes): 30
Depart time (HH:MM): 07:00
Train solution found (1 segment(s)):
Segment 1: Train 10123, from station 1 to 2, depart at 08:00, arrive at 10:30, price: 120.00
```



# 运行结果——航班加入与查询

```
Enter your choice: 5
Enter flight_no: CZ5111
Enter from city: 北京
Enter to city: 昆明
Enter depart time (HH:MM): 10:00
Enter arrive time (HH:MM): 13:00
Enter price (yuan, integer): 1000
Flight added successfully.

== National Transport System ==
1. Add train station
2. Delete train station
3. Add train
4. Delete train
5. Add flight
6. Delete flight
7. Query train (city to city)
8. Query train (station to station)
9. Query flight (city to city)
0. Exit

Enter your choice: 9
Enter start city: 北京
Enter end city: 昆明
Time first(0) or price first(1): 1
Allow transfer? (0/1): 0
Flight solution found (1 segment(s)):
    Flight: CZ5111
Total time: 180 min, total price: 1000.00
```

# 主要代码结构与模块说明

- **main.c**: 主入口, 调用 app 初始化与主循环。
- **app.c/h**: 系统初始化与交互主循环。
- **global.c/h**: 全局变量与初始化。
- **modify.c/h**: 增删查改接口, 自动建站/建城。
- **train.c**: 火车数据导入、最短路与方案输出。
- **plane.c/h**: 航班数据导入、查询与增删。
- **graph.c/h**: 分层图、节点、边、Dijkstra 实现。
- **heap.c/h**: 小根堆, 支持多关键字优先级。

# 总结与体会

- 本系统实现了复杂的交通网络建模与多条件最优路径查询。
- 采用分层图、哈希表、堆等多种数据结构，兼顾效率与灵活性。
- 支持中文输入输出，批量导入，用户体验友好。
- 工程结构复杂，模块划分清晰，便于维护与扩展。
- 收获了大规模数据结构设计与工程实现的宝贵经验。