

# 火车票大作业讨论

---

0516

## 哈希相关：

用户名：哈希到在文件中的位置

车次名：同上

站名：直接哈希

Outline

## 用户：

(≤一万个用户 in 文件01 (user\_btree, user\_info) )

- `username`：用户的唯一标识符，由字母开头，字母、数字和下划线组成的字符串，长度不超过20。
- `password`：由字母、数字和下划线组成的字符串，长度不低于6，不超过30。
- `name`：由二至五个汉字组成的字符串。
- `mailAddr`：同一般邮箱地址格式，且仅含数字、大小写字母，`@` 和 `.`，长度不超过30。（无需检验邮箱格式的合法性）
- `privilege`：所处用户组优先级，为一个0~10的整数。
- `order`：订单信息（具体见下）
  - 在末尾加
  - 在当中删
    - 更改订单状态
    - 调整候补问题

## 订单：（不进内存）

（文件02 (order\_btree) B+树）

- 第一关键字：用户 `username`
- 第二关键字：订单ID `orderID`

（文件03 (order\_info) info\_file 订单信息，即B+树叶节点为订单编号，编号在该文件对应具体信息）

- 订单信息（Order类对象）

相关操作：退票后提上候补（暴力）

## 维护候补信息：

（文件04 (que\_btree) B+树维护）

按照第一、第二关键字检索

- 第一关键字 trainID
  - 第二关键字 发车日期
  - 第三关键字 buy\_time (保证检索后得到有序信息)
  - Value (其他信息) : 见文件05 (userID、orderID)
- (文件05 (que\_info) info\_file文件04的信息)
- userID
  - orderID

### (发布前) 车次:

(文件06 (train\_info\_unreleased) )

- **trainID** : 车次的唯一标识符, 由字母开头, 字母、数字和下划线组成的字符串, 长度不超过20。
  - **stationNum** : 车次经过的车站数量, 为不超过100的整数。
- stations** : 车次经过的所有车站名, 共 **stationNum** 项, 站名由汉字组成, 不超过10个汉字。
- 格式为: **station\_1 station\_2 station\_3 ... station\_stationNum**
  - **seatNum** : 该车次的座位数, 为不超过100000的整数。
- prices** : 每两站之间的票价, 共 **(stationNum - 1)** 项, 第 **i** 项表示 **station\_i** 到 **station\_(i+1)** 的票价, 是一个不超过100000的整数。
- 格式为: **price\_1 price\_2 price\_3 ... price\_(stationNum - 1)**
  - **startTime** : 列车每日的发车时间, 时间格式为 **hr:mi**。如: **23:51**。
  - **travelTimes** : 每两站之间行车所用的时间 (单位为分钟), 共 **(stationNum - 1)** 项, 每一项是一个不超过10000的整数。
  - **stopoverTimes** : 除始发站和终点站之外, 列车在每一站停留的时间 (单位为分钟), 共 **(stationNum - 2)** 项, 每一项是一个不超过10000的整数。
  - **saleDate** : 车次的售卖时间区间 (闭区间), 在时间区间内, 每天都有相同的此车次。共2项, 每一项均为2020年6月至8月的某一天, 日期格式为 **mm-dd**。如: **06-07**。
  - **type** : 列车类型, 为一个大写字母。

可加/删/发布

### 发布前车次索引

(文件07 (train\_id\_unreleased) B+tree)

- 车次ID

### (发布后) 车次:

(文件08 (没用了) )

- 余票信息

(文件09 (train\_info\_released) )

- 车次信息 (见上)

### 发布后车次索引

(文件10 (train\_id\_released) B+tree)

- 车次ID

## 余票信息

(文件11 (ticket\_info) 数组线性存储)

- 发车日期
- 所有余票信息
- 数量级:  $10000 * 100 * 100$
- 存储: in磁盘
- 维护: 每次发车 (车票x售卖区间) 开一个数据结构

query\_ticket

1. 起始站&结束站求交
2. 对日期做核验
3. 时间/价格排序: 前缀和

(文件12 (station\_btree) B+树)

- key: 起始站站名 经过该站的车次
- 见文件13 (value: 该站为该车次的第几站)

(文件13 (station\_info) info\_file 文件12的信息)

- value: train\_id, 该站为该车次的第几站

query\_transfer

1. 获得出发日期+经过始发站的所有可用车次
2. 找到这些车次对应的所有车站
3. 根据这些车站和终点站查票

buy\_ticket & refund\_ticket

1. 修改订单信息
2. 修改余票信息

文件14 (counts):

不存fstream, 每次load的时候开文件、读、关文件, save的时候开文件、写、关文件  
文件内容:

4个int, 分别是user\_cnt,order\_cnt,train\_cnt(unreleased),train\_cnt(released)

注意TrainController里面需要判当前是unreleased还是released

| 文件名                   | fstream位置                                | 1Key 2Value                                     |
|-----------------------|--|---|
| user_btree            | user_controller.btree_file               | pair<int,int><br>User                           |
| user_info             | user_controller.info_file                |   |
| order_btree           | order_controller.btree_file              | pair<pair<int,int>,int><br>Order                |
| order_info            | order_controller.info_file               |   |
| que_btree             | released_train_controller.que_btree_file | tuple<pair<int,int>,Date,int><br>pair<Char,int> |
| que_info              | released_train_controller.que_info_file  |   |
| train_info_unreleased | train_controller_unreleased.info_file    | pair<int,int><br>Train                          |
| train_id_unreleased   | train_controller_unreleased.btree_file   |   |

|                            |                                       |  |
|----------------------------|---------------------------------------|--|
| <b>train_info_released</b> | train_controller_released.info_file   |  |
| <b>train_id_released</b>   | train_controller_released.btree_file  |  |
| <b>ticket_info</b>         | released_train_controller.ticket_file |  |
| <b>station_btree</b>       | ticket_controller.btree_file          | pair<pair<int,int>,pair<int,int>>><br>pair<Char,int> |
| <b>station_info</b>        | ticket_controller.info_file           |  |
| <b>counts</b>              | 不存fstream, 随时读                        |  |

## 0520

完成除了三方交互之外的接口、完成Trie接口

## 0523

完成三方交互接口、完成B+树接口（）