

Problem A: 智能调度

Description :

随着饿了么逐渐成为外卖行业的第一大品牌，为了进一步提高服务品质，即时配送也逐渐成为提高服务的核心环节。如何解决订单的即时配送，通过全局优化来提升用户体验，降低成本是饿了么亟待解决的核心问题。为了解决该问题阿饿君想出了一个智能调度系统，该系统能针对商家产生的订单分配给最合适的骑手，最大化的提升用户体验、降低成本。但是阿饿君天生笨笨的，没法实现该系统，只能求助于聪明的你啦，赶快来拯救可怜的阿饿君吧！

Input :

在本题中，阿饿君会给出 N ($N < 10^3$) 个餐厅午高峰 (10:00-14:00) 的订单详情，具体格式如下：

1. 餐厅信息数据格式

字段	类型	说明
rst_id	整型	餐厅 id
Lng	浮点数	餐厅经度
Lat	浮点数	餐厅纬度

2. 订单信息数据格式

字段	类型	说明
order_id	整型	订单 id
rst_id	整型	餐厅 id
customer_lng	浮点型	客户经度
customer_lat	浮点型	客户纬度
make_order_time	整型	订单出餐时长，单位 s
promised_at	字符串	承诺送达时间 格式：YYYY-MM-DD HH:mm:ss 样例：2016-09-01 10:23:34
created_at	字符串	订单创建时间 格式：YYYY-MM-DD HH:mm:ss 样例：2016-09-01 10:23:34

3. 最多可以使用 M ($M < 10^4$) 个骑手，骑手都有相同的初始位置。

4. 程序一共有三个测试集，按测试集大小分为简单、中等、困难三个层次；程序必须全部通过三个测试机才可以参与排名。

为了帮助你快速实现该调度系统，阿饿君做了如下模型假设：

1. 雇佣一个骑手需要花费 200 元/天；
2. 骑手任何时刻的配送订单总数不能大于 7 单。
3. 骑手去餐厅取餐时，若订单还未出餐骑手需要在该餐厅进行等待；
4. 骑手去客户送餐时，到达客户地址即认为送餐成功，不许额外的时间等待；
5. 当决定雇佣一个骑手是，可以将骑手从初始位置瞬移（不需要耗费任何时间）到指定的任意商家或用户地址，之后骑手只能以 3m/s 的速度在商户、用户之间的直线上行驶。任何两点之间的实际骑行距离 = 1.4 * 球面距离：

a. 两点 (lat1, lng1) 和 (lat2, lng2) 间的球面距离公式：

$$\text{distance} = 2 * R * \arcsin \sqrt{\sin^2 \left(\frac{\pi}{180} * \Delta lat \right) + \cos \left(\frac{\pi}{180} * lat1 \right) * \cos \left(\frac{\pi}{180} * lat2 \right) * \sin^2 \left(\frac{\pi}{180} * \Delta lng \right)}$$

其中， $\Delta lat = \frac{lat1-lat2}{2}$, $\Delta lng = \frac{lng1-lng2}{2}$, R 为地球半径，取 6378137 米。

6. 假设订单的实际送达时间 T，记配送误差为 $\Delta t = \text{Ceil}(\frac{T-promised_at}{60})$, Ceil 为向上取整函数， Δt 单位为分钟。我们会根据配送误差 Δt 对用户进行不同的赔偿；计算公式为一段函数 $f(\Delta t)$ ：

- a. $\Delta t \leq 0$, 赔偿 0 元。
- b. $\Delta t > 0$, 赔偿 $\Delta t * \log(\Delta t + 1) + 5$ 元。

7. 调度方案的总代价为 $\text{total_cost} = 200 * \text{雇佣骑手个数} + \sum_1^N f(\Delta t)$

阿饿君希望聪明的你把这 N 个订单分配给最合适的 m (m <= M) 个骑手来完成这 N 个订单配送，使得总代价尽可能小；

Output:

每个骑手按操作序号升序输出自己的配送操作顺序列表，满足如下格式：

骑手 id，操作序号，骑手经度，骑手纬度，订单 id，操作类型（take/delivery），操作时间

Judge rules:

只要满足下面任一规则条件，即为非法调度方案；非法调度方案不参与最后的排名，记零分。规则如下：

1. 程序接受两个参数输入，第一参数代表 restaurant.csv，第二个参数代表 order.csv。
2. 调度程序必须全部通过三个测试集，否则记零分。
3. 不满足任一模型假设的。
4. 任一订单取餐时间晚于送餐时间。
5. 任一订单的、餐厅、骑手的信息与给定的数据集不一致。
6. 任一订单的取送必须是同一个骑手完成的。
7. 任一骑手同时刻出现在多个不同的位置。
8. 有遗漏订单没有参与调度。
9. 发现类同代码的方案直接取消参赛资格。

Sample Input:

restaurant.csv

1,121.3968400000,31.2145100000

2,121.5841000000,31.2006500000

order.csv

1,1,121.4039800000,31.2061200000,6,2016-09-18 11:11:10,2016-09-18 10:21:10

2,2,121.5878070000,31.2099080000,13,2016-09-18 12:03:40,2016-09-18 11:13:40

3,2,121.5938700000,31.1864400000,7,2016-09-18 11:28:04,2016-09-18 10:38:04

Sample Output :

1,1,121.3968400000,31.2145100000,1,take,2016/9/18 10:21:16

1,2,121.4039800000,31.2061200000,1, delivery,2016/9/18 10:27:41

2,1,121.5841000000,31.2006500000,3,take,2016/9/18 10:38:11

2,2,121.5938700000,31.1864400000,3, delivery,2016/9/18 10:48:22

2,3,121.5841000000,31.2006500000,2,take,2016/9/18 11:13:53

2,4,121.5878070000,31.2099080000,2, delivery,2016/9/18 11:19:56

输出描述：订单由两个骑手完成，一号骑手配送了第一个订单，二号骑手配送了第 2，3 号订单。3 个订单均未超时未产生罚款。最终代价为 $2*200=400$ 。