

赛道 A：“58 到家”家政服务订单分配问题

“58 到家”是“58 同城”旗下高品质、高效率的上门家政服务平台，平台向用户提供家政保洁、保姆、月嫂、搬家、维修等众多生活领域的服务。在家政保洁场景中，用户在平台下单购买服务后，平台会将订单分配给一个保洁阿姨，阿姨接到订单后按照用户指定的服务时间上门，进行保洁服务。平台在将订单分配给一个保洁阿姨时，一方面，为了提高对顾客的服务质量，需要尽量分配服务分较高的阿姨，其中阿姨的服务分是基于阿姨历史订单的评价情况得到，取值为 $[0,1]$ ，值越大越好；另一方面，为了帮助阿姨提高接单量，需要尽量缩短阿姨相邻单之间的通行时间。

每天通过平台进行分配的订单量是巨大的，当前平台实现了一套订单分配算法，本问题研究的是如何优化系统的分配算法，提高算法的求解能力，实现提升顾客体验、节省阿姨时间。

数据说明：

数据包含一天内、一个区域的所有订单和所有保洁阿姨。

表 1：订单数据表字段说明

字段	数据类型	说明
id	int	订单唯一标识
createTime	int	订单下单时间，单位秒
serviceFirstTime	int	服务开始时间的最早时间，单位秒
serviceLastTime	int	服务开始时间的最晚时间，单位秒
serviceUnitTime	int	服务时长，单位分钟
x	int	服务地点横坐标，单位米
y	int	服务地点纵坐标，单位米

示例：

id,createTime,serviceFirstTime,serviceLastTime,serviceUnitTime,x,y
0,1661422500,1662786000,1662789600,90,-4693,-12716

1,1661474400,1662768000,1662771600,60,2176,-17371
 2,1661568300,1662771600,1662775200,60,3940,-11414
 3,1661575800,1662789600,1662793200,90,1791,-16026

表 2：阿姨数据表字段说明

字段	数据类型	说明
id	int	阿姨唯一标识
serviceScore	double	阿姨服务分
x	int	阿姨初始点横坐标，单位米
y	int	阿姨初始点纵坐标，单位米

示例：

id,serviceScore,x,y
 0,0.9998,931,-31576
 1,0.9993,7529,-7145
 2,0.9992,5393,2140
 3,0.999,5982,-3340

基本业务抽象

约束条件及假设：

1. 所有订单都要分配一个且只有一个阿姨；
2. 每个订单需要指定一个服务开始时间，这个时间的取值范围为[最早时间，最晚时间]，且是半点的整数倍；
3. 一个阿姨同时只能服务一个订单；
4. 阿姨需要在每个订单的服务开始时间之前到达客户位置；
5. 阿姨每天开始任务时必须从初始点位置出发；
6. 任意两点的距离为欧式距离；
7. 保洁阿姨的行驶速度为 15 千米/小时。

优化目标：

将每个订单匹配阿姨时，优化的目标是：

1. 所有订单匹配的阿姨的服务分，其平均值 **A** 尽可能大；
2. 最小化每单的平均通行距离 **B**。一个订单的通行距离指的是阿姨从上一个地点到本单地点的距离（欧式距离），其中阿姨第一个订单的通行距离等于从初始点到

第一个订单位置的距离，单位是千米；

3. 最小化阿姨服务订单的平均间隔时间 C 。一个订单的间隔时间指的是，阿姨从上一个单服务结束时刻到本单服务开始时刻的时间间隔，单位是小时，其中阿姨第一个订单的间隔时间设定为 0.5 小时（阿姨首单需要做基本的准备工作，不考虑阿姨从初始点到第一个订单的通行时间）；
4. 总体目标是各个目标的加权和： $\alpha A - \beta B - \gamma C$ ，其中 $\alpha=0.78$ 、 $\beta=0.025$ 、 $\gamma=0.195$ ，得分四舍五入取 6 位小数。目标值越大越好。

初赛问题

问题 1：只考虑离线批量派单模式。附件 1 与附件 2 中分别给出的是一天的所有订单信息与阿姨信息。

- (a) 请设计最优的订单与阿姨匹配算法，将所有订单进行分配，并将求解结果填写到 `result1.txt` 中。（订单必须全部分配、阿姨不需要全部匹配订单）。
- (b) 基于(a)的算法，请对附件 1 中的前 50 个订单与附件 2 中前 20 个阿姨，重新运行算法，给出阿姨的执行任务列表，并画出阿姨的行动轨迹图。

表 3: `result1` 提交结果格式说明

字段	数据类型	说明
id	int	订单唯一标识
serviceStartTime	int	服务开始时间，单位秒
auntId	int	服务阿姨 id

示例：

0,1662786000,260
1,1662768000,61
2,1662771600,84
.....

问题 2：线上批量派单模式。在实际业务场景中，通常采用固定的频率派单，每 30 分钟将该段时间内产生的新订单统一分配；分配时允许部分订单暂时不派单，称之为压单，但是压单订单必须满足服务开始时间的最早时间比当前时间晚于 2 个小时（不包括 2 个小时）也即满足：

serviceFirstTime-currentTime>2h; 请设计这种情况下的每批订单的最优分配算法。并将求解结果 1-最终决策结果填写到 result21.txt 中，结果 2-每次决策结果填写到 result22.txt 中。

表 4：问题 2 结果 1-最终决策结果格式说明

字段	数据类型	说明
id	int	订单唯一标识
serviceStartTime	int	服务开始时间，单位秒
auntId	int	服务阿姨 id

示例：

0,1662786000,260
1,1662768000,61
2,1662771600,84
.....

表 5：问题 2 结果 2-每次决策结果格式说明

字段	数据类型	说明
currentTime	int	当前决策时间，单位秒
id	int	订单唯一标识
serviceStartTime	int	服务开始时间，单位秒，如果压单，则为-1
auntId	int	服务阿姨 id，如果压单，则为-1
retainable	int	是否压单，1 为压单，0 为不压单

示例：

1661423400,0,1662786000,260,0
1661475600,1,1662768000,61,0
1661569200,2,1662771600,84,0
.....

附件

附件 1：订单数据.txt;

附件 2：阿姨数据.txt;

result: result1.txt

result21.txt

result22.txt

要求将所有问题结果写入到 **result** 文件夹下的 3 个文件中，请勿修改文件夹及文件名，并将 **result** 文件夹压缩为 **zip** 格式，将 **result.zip** 提交到竞赛平台。