

2019 Girl Hackathon

2019 年 3 月 14 日

1 Introduction

1. 路径规划：在一个给定的城市地图模型下，给自动快递机优化配送路径。自动快递机需要从配送中心出发，经过若干个客户点，返回配送中心。需要设计算法寻找时间最短的路径。

2. 产品应用：基于 1 中的算法，结合实际问题开发相应的产品，for example: fancy visualization, 或者给快递员使用的 app or website.

2 问题描述

给定的城市是网格形状的，由 N ($1 \leq N \leq 3000$) 条横向的道路和 M ($1 \leq M \leq 3000$) 条纵向的道路组成。道路的交叉点都是安装了红绿灯的路口，相邻两个路口的距离都算一样的。(如图 1 所示)



图 1: Smart City

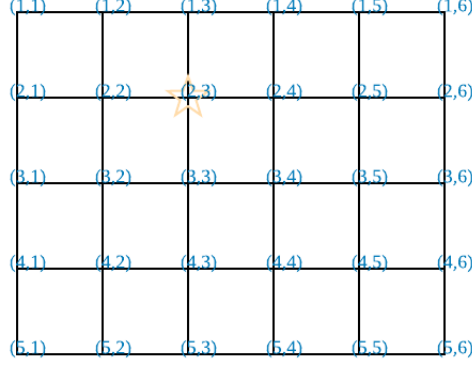


图 2: Sample map for $N = 5, M = 6, s_x = 2, s_y = 3$

我们使用二维坐标来表示每一个路口:横坐标从上到下编号为 $1, 2, \dots, N$, 纵坐标从左到右边编号为 $1, 2, \dots, M$. 配送中心的位置位于 (s_x, s_y) 处。(如图 2 所示)

假设我们有 K ($1 \leq K \leq 60$) 个客户点, 客户点的坐标为 (x_i, y_i) ($1 \leq i \leq K$, 客户点坐标两两不相同, 而且不会和配送中心一样)。自动快递机从 $t = 0$ 时刻出发, 经过 K 个客户点 (对经过顺序不做要求), 完成配送任务, 最后返回配送中心。我们需要寻找一条时间最短的路径。

自动快递机只能按照城市的道路行走, 假设从一个路口 (i, j) 走到相邻的路口 $((i-1, j), (i+1, j), (i, j+1), \text{ or } (i, j-1))$ 需要的时间为 30 秒。每个路口都有红绿灯, 经过路口需要等待红绿灯才能通过。路口 (i, j) 有三个参数: T_{ij}^G, T_{ij}^R 和 t_{ij} ($1 \leq T_{ij}^G, T_{ij}^R, t_{ij} \leq 1000, t_{ij} \leq T_{ij}^G$), 表示该路口绿灯亮的时间长度为 T_{ij}^G 秒, 红灯亮的时间长度为 T_{ij}^R 秒, 通过这个路口需要的时间为 t_{ij} 秒。无论是左转、右转、直行或者掉头, 需要的时间都是 t_{ij} 秒, 而且在通过的 t_{ij} 秒内都是绿灯才能通过, 否则需要等待下一个绿灯。每个路口的红绿灯都是绿灯、红灯循环切换的, 而且在 $t = 0$ 时刻都是刚好开始亮绿灯。

假设配送路径中, 第一次从配送中心出发和最后一次到达配送中心, 不需要花费额外的时间。

3 输入数据格式

第一行是两个用空格分隔的整数 N 和 M ，其中 $1 \leq N, M \leq 3000$ 。

第二行是配送中心的坐标 (s_x, s_y) ， $(1 \leq s_x \leq N, 1 \leq s_y \leq M)$ 。

第三行是客户点的个数 K ， $(1 \leq K \leq 60)$ 。

接下来的 K 行是 K 个客户点的坐标 (x_i, y_i) ($1 \leq i \leq K$)，其中 $1 \leq x_i \leq N, 1 \leq y_i \leq M$ ，而且这 K 个客户点必须两两不相同，而且不能和配送点重合。

接下来的 $N \times M$ 行，每行 5 个整数 $(i, j, T_{ij}^G, T_{ij}^R, t_{ij})$ ($1 \leq i \leq N, 1 \leq j \leq M, 1 \leq T_{ij}^G, T_{ij}^R, t_{ij} \leq 1000, t_{ij} \leq T_{ij}^G$)。这 $N \times M$ 行指出了 $N \times M$ 个路口的三个参数。

4 输出数据格式

第一行是该路径需要花费的时间，单位为秒。从第二行开始，按照顺序输出该路径经过的路口的坐标。（第一个坐标和最后一个坐标必须是配送中心）。

5 样例

输入数据:

```
5 6
2 3
2
3 4
4 1
1 1 8 8 5
1 2 8 8 5
1 3 8 8 5
1 4 8 8 5
1 5 8 8 5
1 6 8 8 5
2 1 8 8 5
2 2 8 8 5
```

2 3 8 8 5
2 4 8 8 5
2 5 8 8 5
2 6 8 8 5
3 1 8 8 5
3 2 8 8 5
3 3 8 8 5
3 4 8 8 5
3 5 8 8 5
3 6 8 8 5
4 1 8 8 5
4 2 8 8 5
4 3 8 8 5
4 4 8 8 5
4 5 8 8 5
4 6 8 8 5
5 1 8 8 5
5 2 8 8 5
5 3 8 8 5
5 4 8 8 5
5 5 8 8 5
5 6 8 8 5

该样例输入的信息如图 3 所示。

一个合法的输出为：

387
2 3
2 2
3 2
3 1
4 1
4 2
3 2
3 3

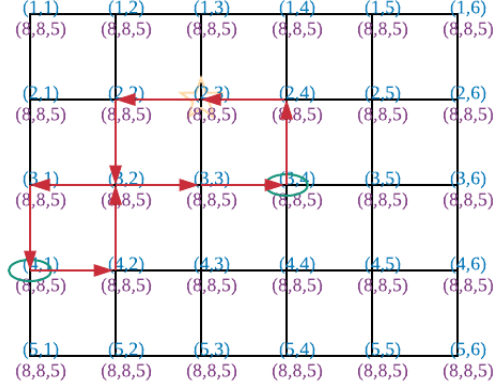


图 3: Example

3 4

2 4

2 3

该路径的详细步骤:

 $t = 0$: leaves (2, 3) $t = 30s$: arrives (2, 2) $t = 32s$ to $t = 37s$: leaves (2, 2), waited 2 seconds $t = 67s$: arrives (3, 2) $t = 67s$ to $t = 72s$: leaves (3, 2) $t = 102s$: arrives (3, 1) $t = 112s$ to $t = 117s$: leaves (3, 1) $t = 147s$: arrives (4, 1) $t = 147s$ to $t = 152s$: leaves (4, 1) $t = 182s$: arrives (4, 2) $t = 192s$ to $t = 197s$: leaves (4, 2) $t = 227s$: arrives (3, 2) $t = 227s$ to $t = 232s$: leaves (3, 2) $t = 262s$: arrives (3, 3) $t = 272s$ to $t = 277s$: leaves (3, 3) $t = 307s$: arrives (3, 4)

$t = 307s$ to $t = 312s$: leaves (3, 4)

$t = 342s$: arrives (2, 4)

$t = 352s$ to $t = 357s$: leaves (2, 4)

$t = 387s$: arrives (2, 3)

6 提交步骤

1. (Sunday 10:00 前) 每个队提交 10 组用于 PK 的数据, 每组数据需要满足输入数据的格式和范围, 文件命名为 XXXX+input_1.txt, XXXX+input_2.txt, ..., XXXX+input_10.txt (XXXX 为 team name)。

2. (Sunday 11:00) 裁判组会合并各个队提供的数据, 去除不合法的数据, 得到 X 组评测数据 (如果所有队伍的数据都是合法的, X 会等于 10 倍的队伍数), 文件命名为 input_1.txt, input_2.txt, ..., input_X.txt, 所有测试文件放入文件名为 input 的文件夹中, 发送给各个队伍。同时提交能够生成该结果的源程序。

各个队伍收到最终 PK 数据集后, 将有 30 分钟的时间去跑出最终的结果。

3. (Sunday 11:30) 各个队伍提交针对每一个输入数据的输出结果, 文件命名为 output_1.txt, output_2.txt, ..., output_X.txt, 放入以队名命名的文件夹中, 提交路径规划结果以及源代码。

路径规划部分的结果评测: 针对每一组测试数据, 将按照每个队给出路径的时间进行从小到大排名, 如果没有给出答案或者路径非法, 排名为最后一名。所有数据的排名总和为该队在算法评测中的总分 (总分越低, 最后排名越靠前)。

4. (Sunday 12:30 to 13:00) 提交产品应用部分的代码和 demo。

7 注意事项

1. 注意提交文件的文件名和格式。
2. 提交的源程序必须在 30 分钟内跑出对应的结果, 否则结果无效。
3. 不得在源程序中对数据进行特判以后直接输出结果, 否则该组数据结果无效。
4. 按时提交对应的结果。