

# 外卖骑手

## 题目描述

小A是一名大学生，由于经常给游戏充值，导致生活费不足，于是决定通过送外卖来补贴生活。

他收到了一批外卖订单，总共有  $n$  条，每条订单包含两个位置：商家位置  $rest_i$  和顾客位置  $cust_i$ ，分别表示取餐和送餐的地点。小A需要先前往商家取餐，再将外卖送到对应的顾客。小A的电动车上可以装无限数量的外卖，任务是找到一个最优路线，使得小A花费的送外卖时间最少。

小A初始在1号点，你需要帮助小A安排最短的送外卖路线，求出他完成所有订单所需的最短时间。

地图以距离矩阵的形式给出，矩阵中的值表示每两个位置之间的距离。若某些位置之间没有道路相连，则用  $-1$  表示。

请注意，**每条订单必须先取餐再送餐，允许取多个餐以后再送餐。**

## 输入格式

第一行包含两个整数  $n$  和  $m$ ，分别表示外卖订单的数量和城市中地点的总数。

接下来是一个  $m \times m$  的距离矩阵，矩阵中的第  $i$  行第  $j$  列表示从地点  $i$  到地点  $j$  之间的行驶时间。若两个地点之间没有直接相连的道路，则值为  $-1$ ，注意这里的道路都是单向道路。

接着是  $n$  行，每行包含两个整数  $rest_i$  和  $cust_i$ ，分别表示第  $i$  条外卖单中商家和顾客的位置。

## 输出格式

输出一个整数，表示完成所有外卖订单的最短时间。

## 输入样例 #1

```
1 | 2 4
2 | 0 2 -1 1
3 | -1 0 3 2
4 | -1 -1 0 1
5 | 6 -1 -1 0
6 | 1 3
7 | 2 4
```

## 输出样例 #1

```
1 | 6
```

解释:

- 小A有两条外卖订单：
  - 第一条订单商家在位置 1，顾客在位置 3。
  - 第二条订单商家在位置 2，顾客在位置 4。
- 距离矩阵表示：
  - 从 1 到 2 的距离为 2。
  - 从 1 到 4 的距离为 1。
  - 从 2 到 3 的距离为 3。
  - 从 2 到 4 的距离为 2。

- 从 3 到 4 的距离为 1。
- 从 4 到 1 的距离为 6。

最短路径可以从初始点 1 号点开始，先到1号和2号地点取餐，然后送到3号和4号地点。路径总长度为  $2 + 3 + 1 = 7$ 。

## 输入样例 #2

```
1 3 9
2 0 7 6 3 -1 -1 -1 -1 -1
3 -1 0 2 -1 -1 -1 -1 -1 -1
4 -1 2 0 -1 -1 4 -1 -1 10
5 -1 5 -1 0 2 -1 -1 -1 -1
6 -1 20 -1 -1 0 -1 -1 -1 -1
7 -1 -1 -1 -1 -1 0 -1 -1 2
8 20 -1 -1 -1 -1 -1 0 10 3
9 -1 -1 -1 3 8 4 -1 0 -1
10 -1 -1 -1 -1 -1 -1 -1 6 0
11 2 3
12 9 8
13 6 5
```

## 输出样例 #2

```
1 26
```

## 数据范围与说明

- 对于50%的数据,  $1 \leq n \leq 3, 1 \leq m \leq 10, 1 \leq dist[i][j] \leq 10^3$
- 对于100%的数据,  $1 \leq n \leq 6, 1 \leq m \leq 100, 1 \leq dist[i][j] \leq 10^6, 1 \leq rest_i, cust_i \leq m$
- 对于100%的数据, **保证每个商家和顾客的位置在送货单中只出现一次**, 也就是不存在一个位置上有多多个顾客/商家, 也不存在一个位置既有顾客也有商家。
- 所有数据保证地图中小A能送完所有外卖, 至于送完外卖出不出得来, 我们就不管了。

## 提示

1. 运用《数据结构与算法》中的知识, 思考如何计算任意两个地点之间的最短距离
2. 可以使用深度优先搜索(递归、回溯)算法, 尝试所有可能的送外卖顺序, 求解最短时间。