

《算法分析与设计》课程设计

订单分配

打车派单场景, 假定有 N 个订单, 待分配给 N 个司机。每个订单在匹配司机前, 会对候选司机进行打分, 打分的结果保存在 $N \times N$ 的矩阵 `score`, 其中 `score[i][j]` 代表订单 i 派给司机 j 的分值。

假定每个订单只能派给一位司机, 司机只能分配到一个订单。求最终的派单结果, 使得匹配的订单和司机的分值累加起来最大, 并且所有订单得到分配。

$1 \leq N \leq 8$

$0 \leq \text{score}[i][j] \leq 1000$

题目保证每组数据的最大分数都是唯一的

样例

样例 1

输入:

```
[[1,2,4],[7,11,16],[37,29,22]]
```

输出:

```
[1,2,0]
```

解释:

标号为 0 的订单给标号为 1 的司机, 获得 `score[0][1] = 2` 分,
标号为 1 的订单给标号为 2 的司机, 获得 `score[1][2] = 16` 分,
标号为 2 的订单给标号为 0 的司机, 获得 `score[2][0] = 37` 分,
所以一共获得了 $2 + 16 + 37 = 55$ 分。

尽量减少恶意软件的传播

在一个节点网络中，当且仅当 $\text{graph}[i][j] = 1$ 时，每个节点 i 会与另一个节点 j 直接连接。

一些节点 **initial** 最初被恶意软件感染。只要两个节点直接连接，并且其中至少有一个节点被恶意软件感染，那么这两个节点都将被恶意软件感染。这种恶意软件的传播会一直持续直到没有更多的节点可以被这种方式感染。

假设 $M(\text{initial})$ 是在恶意软件停止传播之后，整个网络中被恶意软件感染的最终节点数。

我们可以从初始列表中删除一个节点。如果移除这一节点将最小化 $M(\text{initial})$ ，则返回该节点。如果有多个节点满足条件，就返回索引最小的节点。

请注意，如果某个节点已从受感染节点的列表 **initial** 中删除，它以后可能仍然因恶意软件传播而受到感染。

```
1 < graph.length = graph[0].length <= 300
0 <= graph[i][j] == graph[j][i] <= 1
graph[i][i] = 1
1 <= initial.length < graph.length
0 <= initial[i] < graph.length
```

样例

样例 1:

```
输入: graph = [[1,1,0],[1,1,0],[0,0,1]], initial = [0,1]
输出: 0
```

样例 2:

```
输入: graph = [[1,0,0],[0,1,0],[0,0,1]], initial = [0,2]
```

输出: 0

样例 3:

输入: graph = [[1,1,1],[1,1,1],[1,1,1]], initial = [1,2]

输出: 1