

## 作业 4

题目名称	幻灯片	任务调度器	导航仪
代号	Slides	Scheduler	Navigator
分数	30	30	40

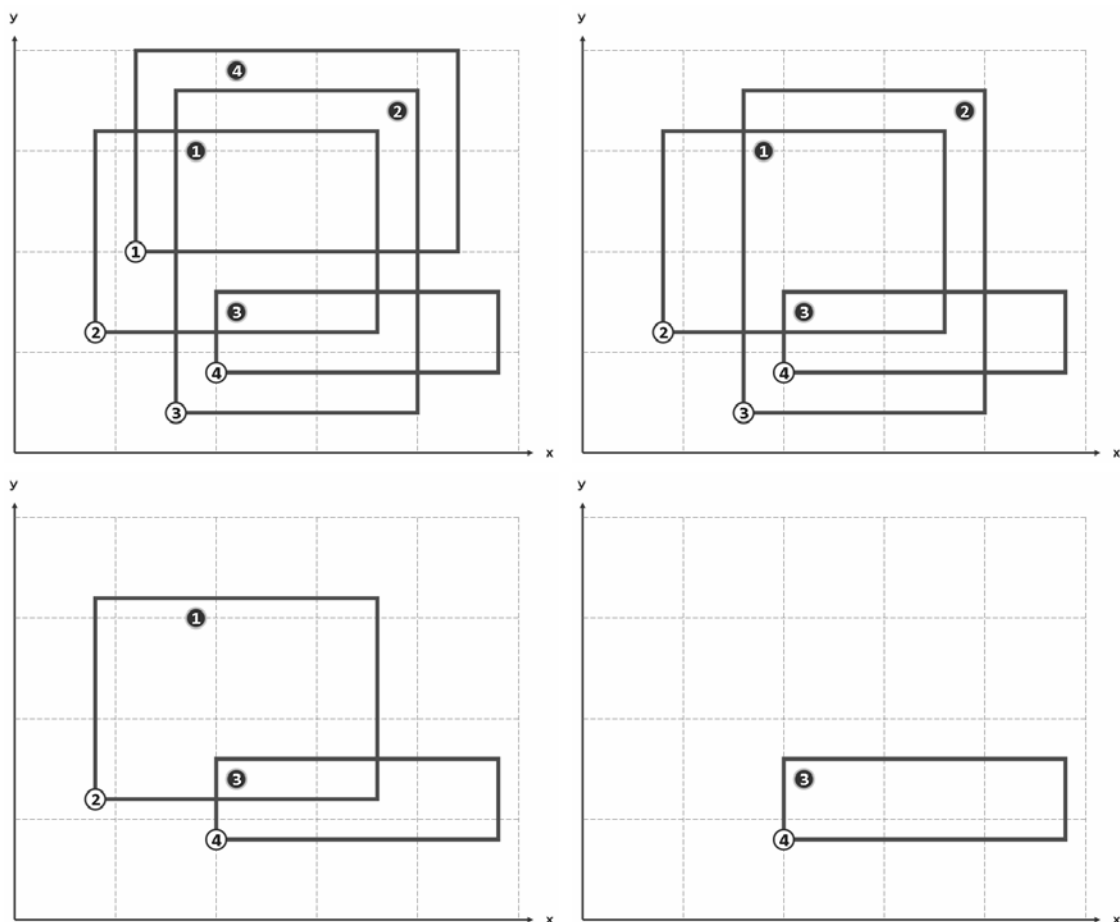
在线测试: <http://166.111.138.40:8888/>

## 第1题 幻灯片 (Slides)

30 分

## 【题目描述】

Sloven 教授平时总是习惯于将演讲所用的幻灯胶片随意堆成一摞，直到最后一刻才请学生代为整理，比如，整理今晚讲稿的任务就交给你了。好在教授已经按播放顺序给每张胶片分别注明了序号 1 ~ n，你拿到胶片后只要按照堆放次序自上而下编号 1 ~ n，则任务的关键就是确定从编号到序号的对应关系。不过，尽管胶片的边都平行于坐标轴，但长宽不一；更令人头疼的是，由于胶片都是透明的，只有落在单张胶片内部的序号才能直接辨识。



在这样的条件下，一个典型的整理过程如图所示，其中黑色圆点为教授标记的序号，白色圆点为你在各胶片左下角标记的编号。首先看到序号 4 只落在编号为 1 的胶片中，可知编号 1 与序号 4 对应，故将这张胶片抽出。此后同理，可依次确定编号 3 与序号 2 对应，编号 2 与序号 1 对应，以及编号 4 与序号 3 对应。这里约定，被胶片边界穿过的序号也视作落在胶片内部。

## 【输入】

第一行为一个整数  $n$ ，即幻灯胶片总数。

接下来的  $n$  行分别对应于编号为 1~n 的胶片，各用 4 个以空格分隔的浮点数  $x_1$ 、 $y_1$ 、 $x_2$  和  $y_2$  给出胶片的左下角和右上角坐标。

再接下来的  $n$  行依次对应于序号  $1 \sim n$ , 各用两个以空格分隔的浮点数  $x$  和  $y$  给出序号所处的坐标。

注意, 这里的输入并不保证按照以上方法, 能够而且唯一确定编号与序号之间的对应关系。具体地, 在如上整理过程的每一步, 对于尚未确定对应关系的序号, 这里并不保证它们的坐标都落在至少一张尚未抽出的胶片内部, 而且至少有一个序号仅落在一张尚未抽出的胶片内部。不难理解, 前一情况意味着教授所标记的序号有误 (比如不小心标记到存放胶片的口袋上); 后一情况则意味着出现歧义, 需借助其它算法方可辨识。无论如何, 这两种情况在这里都称作 “无解”。

**【输出】**

若无解, 只需输出仅含 “-1” 的一行。

否则输出共  $n$  行, 各用两个以空格分隔的整数表示编号与序号的对应关系, 各行应按编号排序。

**【输入样例 1】**

```
4
6.0 10.0 22.0 20.0
4.0 6.0 18.0 16.0
8.0 2.0 20.0 18.0
10.0 4.0 24.0 8.0
9.0 15.0
19.0 17.0
11.0 7.0
11.0 19.0
```

**【输出样例 1】**

```
1 4
2 1
3 2
4 3
```

**【输入样例 2】**

```
2
0.0 0.0 2.0 2.0
0.0 0.0 2.0 2.0
1.0 1.0
1.0 1.0
```

**【输出样例 2】**

```
-1
```

**【限制】**
$$1 \leq n \leq 1,000$$
**【提示】**

拓扑排序

**第2题 任务调度器 (Scheduler)****30 分****【题目描述】**

某高性能计算集群 (HPC cluster) 采用的任务调度器与众不同。为简化起见，假定该集群不支持多任务同时执行，故同一时刻只有单个任务处于执行状态。初始状态下，每个任务都由称作优先级数的一个整数指定优先级，该数值越小优先级越高；若优先级数相等，则任务名 ASCII 字典顺序低者优先。此后，CPU 等资源总是被优先级数最小的任务占用；每一任务计算完毕，再选取优先级数最小下一任务。不过，这里的任务在计算结束后通常并不立即退出，而是将优先级数加倍（加倍计算所需的时间可以忽略）并继续参与调度；只有在优先级数不小于  $2^{32}$  时，才真正退出。

你的任务是，根据初始优先级设置，按照上述调度原则，预测一批计算任务的执行序列。

**【输入】**

第一行为以空格分隔的两个整数  $n$  和  $m$ ， $n$  为初始时的任务总数， $m$  为预测执行序列的长度。

以下  $n$  行分别包含一个整数和一个由不超过 8 个小写字母和数字组成的字符串。前者为任务的初始优先级数，后者为任务名。数字和字符串之间以空格分隔。

**【输出】**

最多  $m$  行，各含一个字符串。按执行次序分别给出执行序列中前  $m$  个任务的名称，若执行序列少于  $m$ ，那么输出调度器的任务处理完毕前的所有任务即可。

**【输入样例】**

```
3 3
1 hello
2 world
10 test
```

**【输出样例】**

```
hello
hello
world
```

**【限制】**
$$1 \leq n \leq 50,000, 1 \leq m \leq 500,000$$
**【提示】**

优先队列

**第3题 导航仪 (Navigator)****40 分****【题目描述】**

自驾旅游已成为当下的一种时尚，但面对不断攀升的油价，有车一族不得不学会精打细算。某车载导航仪公司在调研中发现，各加油站的售价不尽相同，因此旅游路线规划是否合理直接决定油费的开销。于是他们瞄准这个商机，着手在导航仪中加入为用户提示油费开销最低路线的功能。为此，他们将问题抽象描述为图：加油站对应于节点，加油站之间的道路对应于边。为简单起见，假定每次旅行的起点和终点本身都是加油站，油耗线性正比于路程长度而与方向无关。

**【输入】**

第一行为两个整数  $n$  和  $m$ ，表示共有  $n$  个加油站以及  $m$  条联接于其间的公路。

以下  $n$  行各用一个整数  $p_i$  给出第  $i$  个加油站的油价， $1 \leq p_i < 100$ ， $0 \leq i < n$ 。

再以下  $m$  行各用三个整数  $i$ 、 $j$  和  $d_{ij}$  给出加油站  $i$  与  $j$  之间公路的距离（即对应的油耗） $d_{ij}$ ，其中  $0 \leq i < j < n$ ， $0 < d_{ij} < 200$ 。

接下来一行用一个整数  $q$  表示查询次数。

以下  $q$  行各用三个整数  $c$ 、 $s$  和  $t$  ( $1 \leq c \leq 100$ ， $0 \leq s < n$ ， $0 \leq t < n$ ) 表示一次查询：对于油箱容量为  $c$  且初始为空的汽车，计算其从  $s$  号加油站到  $t$  号加油站最经济的路线。

**【输出】**

对于每一查询，输出对应的最低汽油花费；如果不存在这样的路线，那么输出 -1。

**【输入样例】**

```
3 3
20
10
30
0 1 10
1 2 15
0 2 20
2
20 0 2
5 1 2
```

**【输出样例】**

```
350
-1
```

**【限制】**

$1 \leq n \leq 500$ ， $1 \leq m \leq 5,000$ ， $1 \leq q < 1000$

**【提示】**

稀疏图、最短路径