

《数据结构》第4次作业

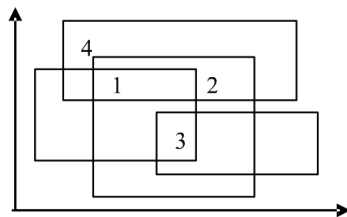
题目一览

名称	代号	满分
幻灯片	slide	30
基站布置	basestation	35
任务调度器	scheduler	35

习题4.1 幻灯片(slide)

【题目描述】

某教授将于今天下午做一个非常重要的演讲。不幸的是他不是一个非常爱整洁的人，他把自己做演讲要用的幻灯片随随便便堆在了一起。因此，演讲之前他不得不去整理这些幻灯片。作为一个讲求效率的学者，他希望尽可能简单地完成它。情况是这样，教授这次演讲一共要用 n 张幻灯片，这 n 张幻灯片按照演讲要使用的顺序已经用数字 $1, 2, \dots, n$ 在上面写下了编号。不过因为幻灯片是透明的，所以我们不能一下子看清每一个数字对应的幻灯片。



如图所示，我们可以很快确定写着的序号为4的幻灯片是哪一张，把它抽出来之后我们又可以确定写着的序号为2的幻灯片是哪一张……

我们用带括号的数字 $(1), (2), \dots, (n)$ 按照幻灯片堆叠的顺序从上往下依次编上号，你的任务是编写一个程序，把幻灯片的数字编号和叠放顺序编号对应起来，显然这种对应应该是唯一的；若是出现多种对应的情况或者某些两种编号对应不起来，我们就称对应是无法实现的。

【输入】

第一行包含一个整数 n ，表示有 n 张幻灯片，接下来的 n 行每行包括4个浮点数 x_1, y_1, x_2, y_2 (数之间用空格隔开)为幻灯片在普通的X-Y坐标系下左下角和右上角的坐标，这 n 张幻灯片按照其在输入中的出现顺序从前往后编号为 $(1), (2), (3) \dots$ 再接下来的 n 行依次为 n 个写在幻灯片上面的数字编号的坐标 x, y ，显然在幻灯片之外是不会有数字的。

【输出】

若是对应可以实现，输出文件一共有 n 行：每一行为一个堆放的顺序编号和对应的幻灯片上面写着的数字编号，中间以一个空格隔开，并且以堆放顺序的编号顺序升序排列；若是对应无法实现，在文件的第一行输出-1即可。行首行末无多余空格。

【样例输入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
```

【样例输出1】

1 4

2 1

3 2

4 3

【样例输入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

【样例输出2】

-1

【限制】

$1 \leq n \leq 1000$

习题4.2 基站布置(basestation)

【题目描述】

在一次军事演习中，某师开始在自己的控制区域内布置通讯基站。出于稳定性考虑，基站间都用有线电缆相连；为了方便布置和维护，电缆都是直接拉直线。然而，演习场的地形比较复杂，有些地方有障碍无法让电缆穿越。

作为一名参谋的你，被委派了策划如何布置通讯基站间的电缆连接的任务。当然，为了节约成本和时间，要求用总长度最短的电缆将所有基站相连。

为了简化问题，不能穿越电缆的障碍区域都描述成一系列直线段，也就是在考虑布置电缆的时候，任何电缆不能和这些直线段相交，但可以重合，也可以相交于这些直线段的某一个端点。

【输入】

第一行包含两个整数 n 和 m ，表示有 n 个基站、 m 个障碍区域。

接下来 n 行，每行有2个带两位小数的浮点数 x_i 、 y_i ，描述的是每个基站在地图上的坐标 (x_i, y_i) 。

接下来 m 行，每行有4个带两位小数的浮点数 x_{i1} 、 y_{i1} 、 x_{i2} 、 y_{i2} ，描述的是表示障碍区域的直线段的两个端点 (x_{i1}, y_{i1}) 、 (x_{i2}, y_{i2}) 。

输入数据保证不会出现无法将所有基站相连的状况。

【输出】

输出电缆的总长度，保留两位小数。

【样例输入】

```
4 1
0.00 0.00
0.00 10.00
10.00 10.00
10.00 -1.00
1.00 -10.00 1.00 9.00
```

【样例输出】

```
31.00
```

【限制】

$1 \leq n \leq 2000$, $1 \leq m \leq 500$

习题 4.3 任务调度器(scheduler)

【题目描述】

在某个集群中，用了一个特别的任务调度方法来调度任务。

每个任务都由一个整数表示优先级，按照从小到大的排序进行调度。但是，任务运行完毕后通常并不直接退出调度队列，而是将优先级数乘以2之后重新放入队列中进行调度；只有在任务的优先级数乘以2之后大于 2^{32} 的时候，这个任务才会从调度队列中删除。这个集群不支持多个任务同时执行，一个时间内只能有一个任务执行。

如果有两个任务的优先级相同，按照任务名的ASCII字典顺序从小到大进行调度。

我们的任务是求出该任务调度器在运行过程中的调度序列。

【输入】

第一行包括两个整数 n 和 m ， n 表示刚开始时队列中的任务数， m 表示要输出的被调度任务的个数。

接下来 n 行，每一行描述一个进程，由一个数字和一个长度不超过8的由小写字母和数字组成的字符串组成。数字描述的是任务开始的时候的优先级，字符串则是任务的名称，数字和字符串之间用1个空格隔开。输入保证不会在输出完 m 个调度的任务前调度队列变成空。

【输出】

m 行，描述的是任务调度的前 m 个任务，按照任务被调度的先后顺序排列。每行描述都是一个字符串，为调度的任务的名称。

【样例输入】

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

【样例输出】

```
hello
hello
world
```

【限制】

$1 \leq n \leq 50,000$ 。 $1 \leq m \leq 500,000$ 。