

2025 山东省队第三轮集训历城站 Contest 4

LeafSeek

2025 年 5 月 29 日

题目名称	象形文字	岩画	万物归零
题目类型	传统题	构造题	传统题
目录	hiero	petro	astro
可执行文件名	hiero	petro	astro
输入文件名	hiero.in	petro.in	astro.in
输出文件名	hiero.out	petro.out	astro.out
测试点时间限制	2 秒	6 秒	6 秒
测试点空间限制	512 MB	512 MB	512 MB
题目总分值	100	100	100
子任务个数	5	4	4
子任务是否等分	否	否	否

A. 象形文字

题目描述

观看电影是一个复杂的过程。你将可观看的电影分为两个列表：

- 第一列表包含 n 部电影。
- 第二列表包含 m 部电影。

因此，你总共有 $n+m$ 部电影可以看。每部电影的评分都是互不相同的整数，取自集合 $\{1, 2, \dots, n+m\}$ 。因为你不希望连续观看太多高分电影或太多低分电影，你设计了以下观看算法：

- 每一步，你从第一列表中选取一部电影观看。之后该电影将从第一列表中消失。
- 你必须交替选择当前可用的最高分电影和最低分电影。第一次选择必须是最高分电影。
- 观看选中的电影后，你将从第二列表中选择另一部电影插入到第一列表的任意位置。当第二列表为空时，你将停止观看电影。注意这意味着第一列表将始终保持恰好 n 部电影。

你希望第一列表最终按评分升序排列。由于你懒得手动处理列表，你不会直接修改列表本身。相反，在算法的每一步，你需要决定插入哪部电影以及插入的位置，以使第一列表变为有序所需的步骤数最少。

输入格式

从文件 `hiero.in` 中读入数据。

输入的第一行包含两个整数 n 和 m 。

第二行包含 n 个整数：第一列表中的电影评分。

第三行包含 m 个整数：第二列表中的电影评分。

保证第一列表和第二列表的评分的并集等于集合 $\{1, 2, \dots, n+m\}$ 。

输出格式

输出到文件 `hiero.out` 中。

输出一行，包含一个整数：使得第一列表有序所需的最少步骤数，如果不可能则输出 -1 。

样例 1

样例 1 输入

```
5 5
3 1 5 2 4
6 8 7 9 10
```

样例 1 输出

```
4
```

样例 1 解释

第一步，你观看评分 5 的电影（当前最高分），然后将第二列表中的 7 插入第一列表的末尾：

- 第一列表 = $[3, 1, 2, 4, 7]$
- 第二列表 = $[6, 8, 9, 10]$

第二步，观看最低分电影 1，插入 8 到末尾：

- 第一列表 = $[3, 2, 4, 7, 8]$
- 第二列表 = $[6, 9, 10]$

第三步，观看最高分电影 8，插入 9 到末尾：

- 第一列表 = $[3, 2, 4, 7, 9]$

- 第二列表 = [6, 10]

第四步，观看最低分电影 2，插入 10 到末尾：

- 第一列表 = [3, 4, 7, 9, 10]
- 第二列表 = [6]

此时第一列表已按升序排列。注意虽然示例中的插入操作都在末尾，但实际可以插入到任意位置。

样例 2 ~ 10

样例 2 ~ 10 输入

见选手目录下的 `hiero/ex_hiero2-10.in`。

样例 2 ~ 10 输出

见选手目录下的 `hiero/ex_hiero2-10.out`。

数据范围与限制

对于所有数据，保证 $1 \leq n \leq 100000$ ， $1 \leq m \leq 200000$ 。

子任务编号	子任务分值	$n \leq$	$m \leq$
1	15	10	10
2	15	50	100
3	15	200	700
4	15	1000	5000
5	40	100000	200000

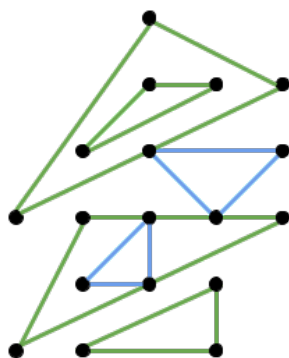
B. 岩画

题目描述

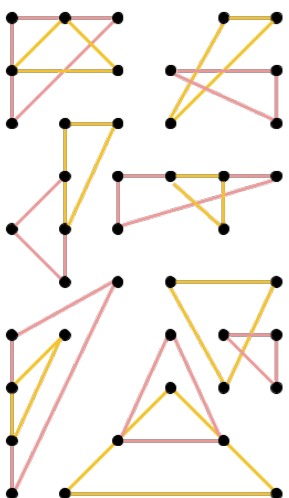
给定一个二维平面上的 n 个互不相同的点构成的集合 P 。你需要找到一个最大的三角形集合，满足以下条件：

1. 集合中的每个三角形的顶点均来自 P ，且每个点最多出现在集合中的一个三角形里。
2. 集合中的每个三角形的面积必须为正（即其三个顶点不共线）。
3. 任意两个三角形的边，它们的交点要么为空，要么是其中一条边的端点。
4. 任意两个三角形的内部区域的交集要么为空，要么等于其中一个三角形（即一个完全包含另一个）。

例如，下图所示的三角形集合满足上述所有条件。



相反，下图中每一对黄色和红色的三角形均不满足条件。



输入格式

从文件 `petro.in` 中读入数据。

第一行输入测试用例的数量 T 。接下来是 T 个测试用例。

每个测试用例的第一行是一个整数 n 。随后 n 行，每行包含两个整数 x_i 和 y_i ，表示第 i 个点的坐标。

输出格式

输出到文件 `petro.out` 中。

对于每个测试用例，输出一行 `Case #x: y`，其中 x 是测试用例编号（从 1 开始）， y 是满足条件的最大三角形集合的大小。

然后，输出 y 行，每行包含三个整数 p_i, q_i, r_i ，表示你构造的第 i 个三角形的顶点是输入中的第 p_i 、 q_i 和 r_i 个点（点的编号从 1 开始）。

样例 1

样例 1 输入

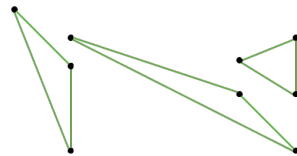
```
3
9
8 2
10 2
2 0
0 5
2 3
10 4
10 0
8 3
2 4
7
0 0
0 3
3 0
0 1
1 0
1 1
2 2
3
0 0
0 1
0 2
```

样例 1 输出

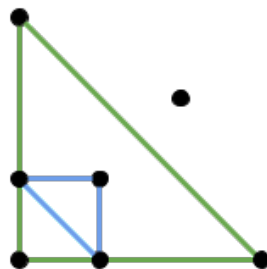
```
Case #1: 3
3 4 5
1 7 9
6 2 8
Case #2: 2
2 3 1
6 5 4
Case #3: 0
```

样例 1 解释

样例 #1 的示意图如下。注意，存在其他有效的构造方式可以达到最大三角形数量。



样例 #2 的示意图如下。同样，存在其他有效的构造方式形成 2 个三角形。



在样例 #3 中，给定的三个点共线，因此无法构成有效的三角形。
注意，输出的三角形顶点顺序可以任意，只要构成有效三角形即可。

样例 2 ~ 5

样例 2 ~ 5 输入

见选手目录下的 petro/ex_petro2-5.in。

样例 2 ~ 5 输出

见选手目录下的 petro/ex_petro2-5.out。

数据范围与提示

对于所有数据，保证 $1 \leq T \leq 100$, $3 \leq n \leq 3000$, $-10^9 \leq x_i, y_i \leq 10^9$ 。对于 $i \neq j$, $(x_i, y_i) \neq (x_j, y_j)$ 。

子任务编号	子任务分值	$n \leq$	特殊性质
1	9	12	存在一条直线 ℓ 穿过 $\geq 2n/3$ 个点
2	21	3000	存在一条直线 ℓ 穿过 $\geq 2n/3$ 个点
3	21	12	无
4	49	3000	无

C. 万物归零

题目描述

给定两个字符串 S 和 T ，下标从 1 开始。

共有 q 次询问，每次询问给定四个整数 a, b, c, d ，要求计算子串 $S[a \dots b]$ 与 $T[c \dots d]$ 的最长公共子序列长度。

输入格式

从文件 `astro.in` 中读入数据。

第一行包含两个正整数 n, m ，分别表示字符串 S 和 T 的长度。

第二行包含一个长度为 n 的小写字母字符串 S 。

第三行包含一个长度为 m 的小写字母字符串 T 。

第四行包含一个正整数 q ，表示询问次数。

接下来 q 行，每行包含四个正整数 a, b, c, d ，满足 $1 \leq a \leq b \leq n, 1 \leq c \leq d \leq m$ 。

输出格式

输出到文件 `astro.out` 中。

输出共 q 行，每行一个整数，第 i 行表示第 i 次询问的答案——子串 $S[a \dots b]$ 与 $T[c \dots d]$ 的最长公共子序列长度。

样例 0

样例 0 输入

```
5 6 7
abaab
babbab
1 5 1 6
1 3 2 4
2 5 2 5
1 4 2 5
2 5 3 6
2 2 5 6
3 4 2 2
```

样例 0 输出

```
4
2
2
3
3
0
1
```

样例 1 ~ 10

样例 1 ~ 10 输入

见选手目录下的 `astro/ex_astro1-10.in`。

样例 1 ~ 10 输出

见选手目录下的 `astro/ex_astro1-10.out`。

数据范围与提示

对于所有数据，保证 $1 \leq n, m \leq 3000$ ， $1 \leq q \leq 10^5$ 。

子任务编号	子任务分值	$n, m \leq$	$q \leq$
1	1	600	600
2	33	600	10^5
3	33	3000	5000
4	33	3000	10^5