

CCF NOI Online 2022

提高级

时间：2022 年 3 月 26 日 8:30 ~ 12:00

题目名称	丹钧战	讨论	如何正确地排序
题目类型	传统型	传统型	传统型
输入文件名	stack.in	discuss.in	sort.in
输出文件名	stack.out	discuss.out	sort.out
每个测试点时限	1.0 秒	1.0 秒	3.0 秒
内存限制	512 MB	512 MB	512 MB
测试点数目	20	10	10
测试点是否等分	是	是	是

提交源程序文件名

对于 C++ 语言	stack.cpp	discuss.cpp	sort.cpp
-----------	-----------	-------------	----------

编译选项

对于 C++ 语言	-lm -O2
-----------	---------

注意事项

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. C/C++ 中函数 main() 的返回值类型必须是 int，程序正常结束时的返回值必须是 0。
3. 提交的程序代码文件的放置位置请参考各省的具体要求。
4. 因违反以上三点而出现的错误或问题，申诉时一律不予受理。
5. 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
6. 程序可使用的栈空间内存限制与题目的内存限制一致。
7. 全国统一评测时采用的机器配置为：Inter(R) Core(TM) i7-8700K CPU @3.70GHz，内存 32GB。上述时限以此配置为准。
8. 只提供 Linux 格式附加样例文件。
9. 评测在当前最新公布的 NOI Linux 下进行，各语言的编译器版本以此为准。

丹钧战 (stack)

【题目描述】

有 n 个二元组 (a_i, b_i) , 编号为 1 到 n 。

有一个初始为空的栈 S , 向其中加入元素 (a_i, b_i) 时, 先不断弹出栈顶元素直至栈空或栈顶元素 (a_j, b_j) 满足 $a_i \neq a_j$ 且 $b_i < b_j$, 然后再将其加入栈中。

如果一个二元组入栈后栈内只有这一个元素, 则称该二元组是“成功的”。

有 q 个询问 $[l_i, r_i]$, 表示若将编号在 $[l_i, r_i]$ 中的二元组按编号从小到大依次入栈, 会有多少个二元组是“成功的”。

询问之间相互独立。

【输入格式】

从文件 `stack.in` 中读入数据。

第一行两个正整数 n, q 。

第二行 n 个正整数表示 a_i 。

第三行 n 个正整数表示 b_i 。

接下来 q 行, 每行两个正整数 l_i, r_i , 表示一组询问。

【输出格式】

输出到文件 `stack.out` 中。

q 行, 每行一个自然数表示一组询问的答案。

【样例输入】

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

【样例输出】

```
3
2
2
3
```

【样例解释】

以第一次询问 $[1, 4]$ 为例。

一开始栈为 $\{\}$ 。

加入 1 号二元组后栈为 $\{(3, 10)\}$ ，栈中只有一个元素，该二元组是“成功的”。

加入 2 号二元组 $(1, 10)$ 时，栈顶的 $(3, 10)$ 的 b 值不大于 2 号二元组的，因此弹栈。此时栈空，2 号二元组入栈，栈为 $\{(1, 10)\}$ ，该二元组是“成功的”。

加入 3 号二元组 $(3, 2)$ ，此时栈顶元素与之 a 值不同， b 值比它更大，因而不需要弹栈，直接将 3 号二元组入栈，栈为 $\{(1, 10), (3, 2)\}$ ，栈中有多个元素，该二元组不是“成功的”。

加入 4 号二元组 $(1, 9)$ ，此时栈顶元素 $(3, 2)$ 的 b 值比它小，弹栈。弹栈后栈顶元素 $(1, 10)$ 与 $(1, 9)$ 的 a 值相同，继续弹栈。此时栈空，4 号二元组入栈，栈为 $\{(1, 9)\}$ ，该二元组是“成功的”。

共有 3 个二元组是“成功的”，因而答案为 3。

【样例 2,3,4】

见选手目录下的 `stack/stack*.in` 与 `stack/stack*.ans`。

【数据范围与提示】

对于所有测试点： $1 \leq n, q \leq 5 \times 10^5$ ， $1 \leq a_i, b_i \leq n$ ， $1 \leq l_i \leq r_i \leq n$ 。

每个测试点的具体限制见下表：

测试点编号	特殊限制
1 ~ 3	$n, q \leq 1000$
4 ~ 6	$n \leq 5000$
7 ~ 10	$n, q \leq 10^5$
11 ~ 12	$b_i = n - i + 1$
13 ~ 15	$a_i = i$
16 ~ 20	无

讨论 (discuss)

【题目描述】

有 n 个人正在打模拟赛，模拟赛有 n 道题目。

有两人都会的题目并且没有人会的题目包含另一个人时，两者之间才会讨论。

(定义第 i 个人会的题目的集合为 S_i ，即当 $S_x \cap S_y \neq \emptyset \wedge S_x \not\subseteq S_y \wedge S_y \not\subseteq S_x$ 时，第 x 人和第 y 人会讨论)

为了让模拟赛的效果更好，希望你可以找出一对会讨论的人或判断不存在。

【输入格式】

从文件 `discuss.in` 中读入数据。

第一行一个正整数 T 表示数据组数，对于每组数据：

第一行一个正整数 n 表示人数和题目数量。

接下来 n 行，第 i 行第一个自然数 k_i 表示第 i 个人会 k_i 道题。接下来 k_i 个正整数，每个数 x 表示第 i 个人会第 x 道题。

【输出格式】

输出到文件 `discuss.out` 中。

对于每组数据：

如果没有会讨论的人，输出 NO。

否则第一行输出 YES，第二行输出两个正整数 x 和 y ，表示第 x 人和第 y 人会讨论。

如果有多种方案，输出任意一种即可。

【样例 1 输入】

```
2
5
4 1 2 3 5
3 1 2 3
2 1 2
1 1
1 4
4
3 1 2 3
3 2 3 4
0
4 1 2 3 4
```

【样例 1 输出】

NO
YES
1 2

【样例 2】

见选手目录下的 *discuss/discuss2.in* 与 *discuss/discuss2.ans*。

【数据范围与提示】

对于所有测试点：令一组数据中 $m = \sum k_i$ ，则 $1 \leq T \leq 5$ ， $1 \leq \sum n \leq 10^6$ ， $1 \leq \sum m \leq 2 \times 10^6$ ， $0 \leq k_i \leq n$ 。

每个测试点的具体限制见下表：

测试点编号	特殊限制
1	$n \leq 300$
2 ~ 3	$n \leq 1000$
4	$n \leq 5000$
5 ~ 7	$n \leq 5 \times 10^4$
8	$k_i \leq 10$
9 ~ 10	无

如何正确地排序 (sort)

【题目描述】

有一个 $m \times n$ 的数组 $a_{i,j}$ 。

定义：

$$f(i, j) = \min_{k=1}^m (a_{k,i} + a_{k,j}) + \max_{k=1}^m (a_{k,i} + a_{k,j})$$

你需要求出 $\sum_{i=1}^n \sum_{j=1}^n f(i, j)$ 。

【输入格式】

从文件 `sort.in` 中读入数据。

第一行两个正整数 m, n 。

接下来 m 行，每行 n 个正整数表示 $a_{i,j}$ 。

【输出格式】

输出到文件 `sort.out` 中。

一行一个正整数，表示答案。

【样例 1 输入】

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

【样例 1 输出】

```
564
```

【样例 1 解释】

以 $f(3, 5)$ 为例:

$$\begin{aligned} f(3, 5) &= \max(a_{1,3} + a_{1,5}, a_{2,3} + a_{2,5}, a_{3,3} + a_{3,5}) + \min(a_{1,3} + a_{1,5}, a_{2,3} + a_{2,5}, a_{3,3} + a_{3,5}) \\ &= \max(9, 7, 10) + \min(9, 7, 10) \\ &= 10 + 7 \\ &= 17 \end{aligned}$$

下面给出 $f(i, j)$ 的数表, 第 i 行第 j 列表示 $f(i, j)$:

20	27	18	22	20
27	34	24	29	23
18	24	20	22	17
22	29	22	24	22
20	23	17	22	18

它们的和是答案 564。

【样例 2,3,4】

见选手目录下的 `sort/sort*.in` 与 `sort/sort*.ans`。

【数据范围与提示】

对于所有测试点:

$$2 \leq m \leq 4, 1 \leq n \leq 2 \times 10^5, 1 \leq a_{i,j} \leq 2 \times 10^5。$$

每个测试点的具体限制见下表:

测试点编号	$m =$	$n \leq$
1	4	3000
2	2	10^5
3 ~ 5	3	10^5
6 ~ 7	4	5×10^4
8 ~ 9		10^5
10		2×10^5