

NOIP2025 模拟赛

HSEFZ 2025.07.01

题目名称	联测缺首	联测缺额	联测缺颈	联测缺尾
题目类型	传统题	传统题	传统题	传统题
目录	study	exam	team	memory
可执行文件名	study	exam	team	memory
输入文件名	study.in	exam.in	team.in	memory.in
输出文件名	study.out	exam.out	team.out	memory.out
提交文件名	study.cpp	exam.cpp	team.cpp	memory.cpp
每个测试点时限	1.0 秒	1.0 秒	2.0 秒	2.0 秒
内存限制	512 MiB	512 MiB	1024 MiB	512 MiB
测试点/子任务数目	5	4	7	5
测试点是否等分	否	否	否	否

注意事项（请仔细阅读）

- 文件名（程序名和输入输出文件名）必须使用英文小写。
- 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。
- 提交的程序代码文件需要放在与题目英语名称相同的子文件夹中。（建议子文件夹内外各放一份）**
- 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）。
- 选手提交的程序源文件必须不大于 100KB。
- 程序可使用的栈空间内存限制与题目的内存限制一致。
- 全国统一评测时采用的机器配置为: Intel(R) Core(TM) i5-9500 CPU @ 3.00GHz，内存 16GB。上述时限以此配置为准。
- 只提供 Linux 格式附加样例文件。
- 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。

联测缺首 (study)

题目描述

在高一的第二个学期的 M 个星期的时间内，有 N 门课，编号为 $1 \sim N$ 。每个星期有 N 个课时，第 i 个课时上课程 i 的一节课。

萧瑞希是一位高一学生。对于 $N \times M$ 个课时中的每一个，他会选择如下行动中的一个：

- 行动 1：萧瑞希去上课。如果他去上了课程 i 的一节课，那么他对课程 i 的理解程度会增加 A_i 。
- 行动 2：萧瑞希不去上课。他转而选择任意一门课，并且自学选中的那门课。如果他选中了课程 i 进行了时长为一课时的自学，那么他对课程 i 的理解程度会增加 B_i 。

一开始，对每门课的理解程度都为 0。由于萧瑞希想要在课后练习算法竞赛，他在非上课时间内不会学习。当第三个学期的所有课时结束后，期末考就会举行。

萧瑞希不想挂科。所以他想要最大化在期末考时对每门课的理解程度的最小值。

给定学期的长度，课程的数量，以及对理解程度的提升数值，请写一个程序计算在期末考时对每门课的理解程度的最小值的最大可能值。

输入格式

从文件 `study.in` 读入数据。

第一行，两个正整数 N, M 。

第二行， N 个正整数 A_1, A_2, \dots, A_N 。

第三行， N 个正整数 B_1, B_2, \dots, B_N 。

输出格式

输出到文件 `study.out` 中。

输出一行，一个数，表示在期末考时对每门课的理解程度的最小值的最大可能值。

样例 1 输入

```
3 3
19 4 5
2 6 2
```

样例 1 输出

```
18
```

样例 1 解释

举个例子，如果萧瑞希按如下方式学习，则他对课程 1, 2, 3 的理解程度将分别为 19, 18, 19。

- 第一周课程 1 的课：自学课程 2；
- 第一周课程 2 的课：自学课程 2；
- 第一周课程 3 的课：去上课程 3 的课；
- 第二周课程 1 的课：去上课程 1 的课；
- 第二周课程 2 的课：自学课程 3；
- 第二周课程 3 的课：去上课程 3 的课；
- 第三周课程 1 的课：自学课程 3；
- 第三周课程 2 的课：自学课程 2；
- 第三周课程 3 的课：去上课程 3 的课。

由于对每门课的最小的理解程度不能大于等于 19，输出 18。

该组样例满足数据范围中的子任务 3, 5 的限制。

样例 2 输入

```
2 1
9 7
2 6
```

样例 2 输出

```
7
```

样例 2 解释

该组样例满足数据范围中的子任务 1, 3, 5 的限制。

样例 3 输入

```
5 60000
630510219 369411957 874325200 990002527 567203997
438920902 634940661 593780254 315929832 420627496
```

样例 3 输出

```
41397427274960
```

样例 3 解释

该组样例满足数据范围中的子任务 3, 5 的限制。

样例 4 输入

```
4 25
1 2 3 4
1 2 3 4
```

样例 4 输出

```
48
```

样例 4 解释

该组样例满足数据范围中的子任务 2, 3, 4, 5 的限制。

样例 5

见选手目录下的 `study/study5.in` 与 `study/study5.ans` 。

该组样例满足数据范围中的子任务 1 的限制。

样例 6

见选手目录下的 `study/study6.in` 与 `study/study6.ans` 。

该组样例满足数据范围中的子任务 2 的限制。

样例 7

见选手目录下的 `study/study7.in` 与 `study/study7.ans` 。

该组样例满足数据范围中的子任务 3 的限制。

样例 8

见选手目录下的 `study/study8.in` 与 `study/study8.ans` 。

该组样例满足数据范围中的子任务 4 的限制。

样例 9

见选手目录下的 `study/study9.in` 与 `study/study9.ans` 。

该组样例满足数据范围中的子任务 5 的限制。

数据范围

对于所有数据，

- $1 \leq N \leq 3 \times 10^5$;
- $1 \leq M \leq 10^9$;
- $\forall 1 \leq i \leq N, 1 \leq A_i, B_i \leq 10^9$ 。

子任务编号	特殊限制	分值
1	$M = 1$	10

子任务编号	特殊限制	分值
2	$N \cdot M \leq 3 \times 10^5, A_i = B_i$	25
3	$N \cdot M \leq 3 \times 10^5$	25
4	$A_i = B_i$	25
5	无	15

联测缺额 (exam)

题目描述

华二信息组学生们集思广益，想出了 n 种题目，第 i 种题目中有 a_i 个变量，其中只有 b_i 个跟其答案有关的关键变量。每种题目都有 m 个版本可用，版本之间只有变量取值的区别，所以它们的 a_i 和 b_i 都是完全一样的。于是题库里现在有 $n \times m$ 道题。

一位组题经验的同学提出，题目之间可以强行缝合：将任意两道题目缝合后，会得到一道新的题目，其变量数为原来的两题的变量数之和，其关键变量数为原来的两题的关键变量数之和。（注意，是任意两道题目都可以缝合，包括同一种题目的不同版本，以及缝合得到的题目都可以用于缝合。）缝合之后原来的两个题会消失，新的题加入题库。最终的试卷将从题库中选择一些题目产生。

对于一道题目，其爽度定义为其关键变量数除以其变量数的值。有两个整数序列 $p_0 = 0, p_1, p_2, \dots, p_k = 100$ 与 v_1, v_2, \dots, v_k 。通过调研，同学们对一道题目的评判标准如下：如果一道题的爽度 $\in (\frac{p_{i-1}}{100}, \frac{p_i}{100}]$ ，那么它的开心度为其变量数乘以 v_i 。可以证明对于任意一道题目存在恰好一个 i 满足要求。

根据试卷规范，一场模拟赛的总变量数不能超过 m 个。组题人想知道，在任意次缝合操作后，选出题库中若干道题目作为联测，所有题目开心度的最大总和。

输入格式

从文件 `exam.in` 读入数据。

第一行包含三个正整数 n, m, k 。

第二行包含 k 个正整数 p_1, p_2, \dots, p_k 。

第三行包含 k 个正整数 v_1, v_2, \dots, v_k 。

接下来 n 行，每行两个非负整数 a_i, b_i 。

输出格式

输出到文件 `exam.out` 中。

输出一行一个非负整数，表示最大总开心度。

样例 1 输入

```
2 10 10
10 20 30 40 50 60 70 80 90 100
1 1 1 1 1 3 3 3 4 5
4 2
6 5
```

样例 1 输出

```
30
```

样例 1 解释

两个题缝合起来形成一道 10 个变量，7 个关键变量的题，对应的 $v = 3$ ，总开心度为 $10 \times 3 = 30$ 。

样例 2 输入

```
3 13 4
34 67 99 100
1 10 10 20
7 7
3 2
3 1
```

样例 2 输出

```
200
```

样例 2 解释

一种可行的答案：一道 7 个变量，7 个关键变量的题加上一道 6 个变量，3 个关键变量的题（这个题是合并而来的）。总开心度为 $7 \times 20 + 6 \times 10 = 200$ 。注意这并不是唯一的最优解。

样例 3 输入

```
2 13 2
50 100
1 100
3 2
13 6
```

样例 3 输出

```
1200
```

样例 4

见选手目录下的 `exam/exam4.in` 与 `exam/exam4.ans` 。

该组样例满足数据范围中的子任务 1 的限制。

样例 5

见选手目录下的 `exam/exam5.in` 与 `exam/exam5.ans` 。

该组样例满足数据范围中的子任务 2 的限制。

样例 6

见选手目录下的 `exam/exam6.in` 与 `exam/exam6.ans` 。

该组样例满足数据范围中的子任务 3 的限制。

样例 7

见选手目录下的 `exam/exam7.in` 与 `exam/exam7.ans` 。

该组样例满足数据范围中的子任务 4 的限制。

数据范围

对于所有数据，

- $1 \leq n, m \leq 8000$;
- $1 \leq k \leq 100$;
- $\forall 1 \leq i \leq n, 0 < b_i \leq a_i \leq 8000$;
- $\forall 1 \leq i \leq k, 1 \leq v_i \leq 10^5$, 且 $\forall 2 \leq i \leq k, v_{i-1} \leq v_i$ 。
- $\forall 1 \leq i \leq k, 1 \leq p_i \leq 100, p_{i-1} < p_i, p_k = 100$ 。

子任务编号	$n, m \leq$	特殊性质	分值
1	4	无	10
2	400	无	30
3	8000	A	20
4	8000	无	40

特殊性质 A：所有 v 相同。

联测缺颈（team）

题目描述

Yuna 有 n 个小伙伴，她决定从中选出 3 个小伙伴组成一个队伍帮她打游戏。每个小伙伴的游戏角色有三项能力值：攻击力、防御力、魔力。能力值越高说明该能力越强。

一个队伍的总能力定义为：三人中攻击力最强的人的攻击力值、三人中防御力最强的人的防御力值、三人中魔力最强的人的魔力值 之和。此外，为了使分工明确，Yuna 希望她选出的队伍满足每个成员都有自己的特长，即：每个成员都有一项能力值严格大于其他两人的对应能力值。

在所有符合条件的组队中，Yuna 想要选一个总能力最强的队伍，请你帮她计算出最大的总能力值。若不存在符合条件的队伍，输出 `-1`。

输入格式

从文件 `team.in` 读入数据。

第一行一个整数 n 表示小伙伴的总人数。

接下来 n 行每行三个正整数 a_i, b_i, c_i 分别表示第 i 个小伙伴的攻击力、防御力和魔力。

输出格式

输出到文件 `team.out` 中。

输出一行一个整数表示队伍总能力的最大值。如果不存在符合条件的组队，输出 `-1`。

样例 1 输入

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

样例 1 输出

```
13
```

样例 1 解释

由伙伴 1, 4, 5 组成的队伍符合条件，因为：

- 伙伴 1 的特长是魔力。
- 伙伴 4 的特长是防御力。
- 伙伴 5 的特长是攻击力。

总能力值为： $5 + 4 + 4 = 13$ 。

可以证明这是符合条件的组队中，总能力值最高的队伍。

注意如果选择伙伴 1, 3, 5，总能力值将达到 15，但是这会导致伙伴 1 没有特长，不符合条件。

样例 2 输入

```
8
1 1 1
1 1 5
1 5 1
5 1 1
1 5 5
5 1 5
5 5 1
5 5 5
```

样例 2 输出

```
15
```

样例 3 输入

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

样例 3 输出

```
-1
```

样例 4

见选手目录下的 `team/team4.in` 与 `team/team4.ans`。

该组样例满足数据范围中的子任务 1 的限制。

样例 5

见选手目录下的 `team/team5.in` 与 `team/team5.ans`。

该组样例满足数据范围中的子任务 3 的限制。

样例 6

见选手目录下的 `team/team6.in` 与 `team/team6.ans`。

该组样例满足数据范围中的子任务 7 的限制。

数据范围

对于所有数据，满足：

- $3 \leq n \leq 1.5 \times 10^5$;
- 对于所有 $1 \leq i \leq n$, $1 \leq a_i, b_i, c_i \leq 10^8$ 。

子任务编号	$n \leq$	$a_i, b_i, c_i \leq$	分值
1	300	10^8	8
2	4000	10^8	29
3	1.5×10^5	5	9
4	1.5×10^5	20	9
5	1.5×10^5	300	9
6	1.5×10^5	4000	9
7	1.5×10^5	10^8	27

联测缺尾 (memory)

题目描述

你需要维护 n 个可重集合，初始为空，有 m 中操作：

- 操作 1：区间 $[l, r]$ 的集合中都插入 k 。
- 操作 2：设 T 为区间 $l \sim r$ 的集合的并集中最大的数，则对于所有 $k \in [l, r]$ ，若集合 k 中存在数 T ，集合 k 中删除恰好一个 T 。
- 操作 3：设 T 为区间 $l \sim r$ 的集合的并集中最大的数，查询 T ，若并集为空输出 -1 。

输入格式

从文件 `memory.in` 读入数据。

第一行两个数 n, m 。

接下来 m 行，第一个数代表操作种类 op ：

- 对于操作 1，接下来还有三个数 l, r, k ；
- 对于操作 2 或 3，接下来还有两个数 l, r 。

输出格式

输出到文件 `memory.out` 中。

对于每个操作 3 输出一行，代表答案。

样例 1 输入

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

样例 1 输出

3

样例 1 解释

第一次操作： $\{2\}, \{2\}, \{2\}, \emptyset, \emptyset$
第二次操作： $\{2\}, \{2, 3\}, \{2, 3\}, \{3\}, \emptyset$
第三次操作： $\{2\}, \{2, 3\}, \{2\}, \{3\}, \emptyset$
第四次操作询问 区间 $[1, 3]$ 的最大值，所以答案是 3。

样例 2 输入

```
6 6
1 1 6 2
1 3 3 2
1 3 4 3
2 3 4
3 3 3
3 4 4
```

样例 2 输出

2
2

样例 2 解释

第一次操作： $\{2\}, \{2\}, \{2\}, \{2\}, \{2\}, \{2\}$
第二次操作： $\{2\}, \{2\}, \{2, 2\}, \{2\}, \{2\}, \{2\}$
第三次操作： $\{2\}, \{2\}, \{2, 2, 3\}, \{2, 3\}, \{2\}, \{2\}$
第四次操作： $\{2\}, \{2\}, \{2, 2\}, \{2\}, \{2\}, \{2\}$
第五次操作询问 3 的最大值，所以答案是 2。
第六次操作询问 4 的最大值，所以答案是 2。

数据范围

对于所有数据，满足：

- $1 \leq n, m \leq 2 \times 10^5;$
- $1 \leq k \leq 10^9。$

子任务编号	$n \leq$	$m \leq$	特殊性质	分值
1	10^3	10^3	无	10
2	5×10^4	5×10^4	A	20
3	5×10^4	5×10^4	B	20
4	5×10^4	5×10^4	无	20
5	2×10^5	2×10^5	无	30

特殊性质 A：没有操作 2。

特殊性质 B：操作 2 中 $l = r。$