

OI 训练赛 Day1

[算法基础]

题目名称	英文名	时间限制	内存限制	答案比较方式
伐木	tree	1s	128MB	全文比较，忽略行尾空白和文末回车
数字表格	num	1s	128MB	全文比较，忽略行尾空白和文末回车
剑客	sword	1s	128MB	全文比较，忽略行尾空白和文末回车
序列	seq	1s	128MB	实数比较

欢迎参加今天的训练赛！本次考试注意事项如下：

- 评测在 64 位 Linux 下使用 lemon 进行，栈空间不限制。
- 每道题目的源代码命名为“题目英文名.cpp”，输入文件为“题目英文名.in”，输出文件为“题目英文名.out”。
- 所有题目均需要建立子文件夹，文件目录与 NOIP 一致。这也就是说：您需要提交一个文件夹，这个文件夹中包含几个子文件夹，分别命名为题目英文名。在每个子文件夹下，需要有一份源码，命名为“题目英文名.cpp”。
- 如果对上述的规则有疑问，请直接询问讲师。答疑范围与 NOIP 一致：考场上只回答与具体题目内容无关的询问，或是解释模糊不清的题意。
- 建议在考试结束前 15 分钟停止编码，检查每一题的输入输出、所开的内存空间大小，以免翻车。
- 题目难度未必是递增的，个人擅长的方向也有所不同，故建议不要死磕某一题。智慧人生，品味舍得。
- 友谊第一，比赛第二，无需计较比赛结果。能从比赛中锻炼手感、学点知识，就是好的 ^_^

那么，祝君好运！

Good luck, Have fun!

阮行止, rxz@luogu.org

A. 伐木(tree)

题目描述

“ 伐木丁丁，鸟鸣嚶嚶。
 出自幽谷，迁于乔木。
 嚶其鸣矣，求其友声。 ”
 ——《诗经·小雅》

森林里有 n 棵树，你的任务是把它们全都砍掉。每棵树有自己的高度，而我们每次砍树，都是砍掉树顶端的一截。一棵树的高度被我们砍到 0 时，我们认为这棵树已经砍完了。

我们知道，能否制造和使用工具，是人类与其他动物的根本区别。作为聪明的人类，你手中有一台伐木机，它的功能是：每次可以任选若干棵树，再任选一个长度 k ，伐木机把这些树都砍掉 k 的长度。当然了，树不能被砍到负数高度——这意味着 k 不能大于你选择的树中的最低高度。

在过去的日子里，森林里的树是每年种一次的。所以，目前森林里这 n 棵树的高度分别为 $1, 2, 3 \dots n$ 。

我们想知道，最少用多少次伐木机，即可砍完这片森林。

输入格式

输入仅一行，一个正整数，表示 n 。

输出格式

输出仅一行，一个正整数，表示至少需要用多少次伐木机。

样例数据

tree.in	tree.out
1	1
2	2
3	2

第二组数据，先利用伐木机把 2 砍掉一米。在这之后，两棵树的高度都是 1 了，我们可以一次性将它们都砍掉。

第三组数据，先将 1 和 3 砍掉一米，树木高度变成 0,2,2。接下来将后两棵树砍掉即可。只使用了两次伐木机。

数据规模与约定

对于前 30% 的数据，保证 $n \leq 10$ 。

对于前 60% 的数据，保证 $n \leq 10000$ 。

对于 100% 的数据，保证 $n \leq 10^{18}$ 。

B. 数字表格(num)

题目描述

许多年后，面对着一代又一代的 OIer，奥雷连诺上校都会回想起那个下午，他的父亲带他去见识数字表格。

这个表格的大小是 $n \times m$ ，也就是说有 n 行， m 列。这不是一张普通的表格；它的每一个元素，都是有意义的：位于第 i 行、第 j 列的数字，恰好是 $i \times j$ 。

那时，奥雷连诺痴迷于一个问题：这个表格中，第 k 大的数字是多少？幼年的奥雷连诺可以记下表上的每一个数，然后给它们从小到大排序，最后数出第 k 大的数。

时光如白驹过隙，匆匆而已……奥雷连诺上校心血来潮，想再玩一次少年时的游戏。然而他的记忆力已经大不如前，只能询问带着计算机的你了。

输入格式

输入仅一行，三个正整数，分别表示 n, m, k 。

输出格式

输出仅一行，一个正整数，表示这个表格中第 k 大的数。

样例数据

num.in	num.out
2 2 2	2
2 3 4	3
1 10 5	5

第二组数据中，表格里有 1、2、2、3、4、6 这六个数，其中第四大的是 3。

数据规模与约定

对于前 40% 的数据，保证 $n, m \leq 1000$ 。

对于前 60% 的数据，保证 $n, m \leq 10000$ 。

对于 100% 的数据，保证 $n, m \leq 500000$, $k \leq n \cdot m$ 。

C. 剑客(sword)

题目描述

每一个人都可能是剑客，譬如李白，十五好剑术，遍干诸侯。

我们的剑客正在攻击一个目标，他需要给目标造成 n 点伤害。

本来，剑客每 x 秒可以打出 1 点伤害，但他可以花费一些金币，来学习一些技能。共有 m 种技能，第 i 个技能耗费掉 b_i 枚金币，可以将剑客的攻击间隔从 x 改为 a_i 。请注意，剑客只能学习至多一种技能。

由于这场战斗是最终决战，现场有 k 名吃瓜群众。剑客可以雇佣任意一名吃瓜群众，第 i 名吃瓜群众可以直接对目标造成 c_i 点伤害，而雇佣这名吃瓜群众需要 d_i 枚金币。注意剑客只能雇佣至多一名吃瓜群众。

剑客有 s 枚金币。现在需要知道：他对目标造成 n 点伤害，最少需要多少秒。

输入格式

第一行，三个正整数，分别表示 n, m, k 。
 第二行，两个正整数，分别表示 x, s 。
 第三行， m 个正整数，表示 a_1, a_2, \dots, a_m 。
 第四行， m 个正整数，表示 b_1, b_2, \dots, b_m 。
 第五行， k 个正整数，表示 c_1, c_2, \dots, c_k 。
 第六行， k 个正整数，表示 d_1, d_2, \dots, d_k 。

输出格式

输出仅一行，一个正整数，表示最短时间。

样例数据

sword.in	sword.out
20 3 2 10 99 2 4 3 20 10 40 4 15 10 80	20
20 3 2 10 99 2 4 3 200 100 400 4 15 100 800	200

第一组数据，我们花费 10 个金币把 x 降到 4，花费 80 个金币雇佣第二个吃瓜群众。吃瓜群众帮我们造成了 15 点伤害，剑客只需要打出 5 点伤害即可。用时 20 秒。

第二组数据，剑客啥也买不起，只能硬打出 20 点伤害，用时 200 秒。

数据规模与约定

首先，我们保证输入数据中的所有整数，都不超过 2×10^9 。另外，每个技能不会使得 x 升高（即 $a_i \leq x$ ）；没有任何吃瓜群众可以造成多于 n 的伤害（即 $c_i \leq n$ ）。

对于前 30% 的数据，保证 $m, k \leq 1000$ 。

对于 100% 的数据，保证 $n, m, k \leq 100000$ 。

D. 序列(seq)

题目描述

考虑一个序列，定义它的“不稳定值”为其所有子区间的“平衡度量”的最大值。
 对于一个子区间，定义它的“平衡度量”为：这个子区间中所有数的和的绝对值。

今给定长为 n 的整数序列 $\{a\}$ 。需要选择实数 x ，使得序列 $a_1 - x, a_2 - x, \dots, a_n - x$ 的“不稳定值”最小。求这个最小的不稳定值。

输入格式

第一行，一个正整数，表示 n 。
 第二行， n 个整数，表示序列 $\{a\}$ 。

输出格式

输出仅一行，一个实数，表示新序列 $\{a_i - x\}$ 可能取到的最小的“不稳定值”。

请注意，本题采用实数比较模式来判断选手答案是否正确。
 只要选手程序的输出与参考答案的误差不超过 10^{-6} ，即判定为答案正确。

样例数据

seq.in	seq.out
3 1 2 3	1.0000000000000000
4 1 2 3 4	2.0000000000000000
10 1 10 2 9 3 8 4 7 5 6	4.5000000000000000

第一组数据，取 $x = 2$ ，序列变为 $-1, 0, 1$ 。此时，序列的“不稳定值”是 1。

第二组数据，取 $x = 2.5$ ，序列变成 $-1.5, -0.5, 0.5, 1.5$ 。此时，平衡度量的最大值为前两个元素造成的 $|-1.5 - 0.5| = 2$ ，或后两个元素造成的 $|0.5 + 1.5| = 2$ 。无论如何，平衡度量的最大值是 2，即为整个序列的“不稳定值”。

数据规模与约定

对于前 40% 的数据，保证 $n \leq 5, |a_i| \leq 5$ 。
 对于 100% 的数据，保证 $n \leq 200000, |a_i| \leq 10000$ 。