

模拟赛2024-6-1

题目名称	修建竹林	选择卡牌	方形的零
目录	bamboo	card	zero
可执行文件名	bamboo	card	zero
输入文件名	bamboo.in	card.in	zero.in
输出文件名	bamboo.out	card.out	zero.out
每个测试点时限	1.0s	2.0s	2.0s
内存限制	512MB	512MB	512MB
试题总分	100	100	100
测试点数目	20	20	20
每个测试点分值	5	5	5
是否有部分分	否	否	否
题目类型	传统型	传统型	传统型

提交的源程序文件名:

对于 C++ 语言	bamboo.cpp	card.cpp	zero.cpp
对于 C 语言	bamboo.c	card.c	zero.c
对于 Pascal 语言	bamboo.pas	card.pas	zero.pas

编译开关:

对于 C++ 语言	-O2 -std=c++11	-O2 -std=c++11	-O2 -std=c++11
对于 C 语言	-O2 -std=c11	-O2 -std=c11	-O2 -std=c11
对于 Pascal 语言	-O2	-O2	-O2

1 修建竹林 (bamboo)

1.1 题目描述

小 D 准备在家门口种一片竹林，竹林中有 N 棵竹子从左到右排成一排。

第一天开始时，所有的竹子高度都为 0。在每一天结束时，所有竹子会长高 1 个单位长度。

对于每一棵竹子，小 D 可以在某一天结束后对其进行一次裁剪操作。裁剪操作完成后，这棵竹子在未来不会再继续长高。小 D 可以在同一天对多棵竹子进行裁剪操作。

由于小 D 实在是太懒了，它每 k 天才会去一趟竹林进行裁剪操作。在小 D 的设想中，第 i 棵竹子的高度为至少为 h_i ，同时也不能太高。因此，小 D 定义了不美观程度：设竹子被裁剪后的高度为 t_1, \dots, t_N ，整片竹林的不美观程度为：

$$\sum_{i=1}^N t_i - h_i.$$

小 D 想在每棵竹子的高度都不低于最低高度 h_i 的同时，满足不美观程度小于 M 。

偷懒的小 D 想知道，在上述情况下， k 可行的最大值是多少。

1.2 输入格式

从文件 bamboo.in 中读取数据。

第一行两个非负整数 N, M ，表示竹子的种类数和不美观程度的上限；

第二行 N 个正整数 h_1, \dots, h_N ，表示每棵竹子的最低高度。

1.3 输出格式

输出到文件 bamboo.out 中。

一行一个整数，表示 k 可行的最大值。

1.4 样例输入 1

```
3 4
1 3 5
```

1.5 样例输出 1

```
3
```

1.6 样例解释

当 k 为 3 时, 小 D 会在第 3 天和第 6 天结束时对竹林进行裁剪。

第 3 天时裁剪第 1, 2 棵竹子, 第 6 天时裁剪第 3 棵竹子, 竹子被裁剪后的高度为 3, 3, 6。

不美观程度之和为 $(3 - 1) + (3 - 3) + (6 - 5) = 3$ 。

可以证明 $k = 3$ 是可行的最大值。

1.7 样例输入 2

```
3 40
10 30 50
```

1.8 样例输出 2

```
32
```

1.9 数据范围和约定

对于前 30% 的数据, $N \leq 10, M \leq 1000, h_i \leq 100$;

对于前 50% 的数据, $M \leq 10^7, h_i \leq 10^6$;

对于另外 20% 的数据, $N \leq 20$;

对于 100% 的数据, $N \leq 100, M \leq 10^{11}, 1 \leq h_i \leq 10^9$ 。

2 选择卡牌 (card)

2.1 题目描述

小 D 和小 B 在玩卡牌游戏，每张卡牌上有三个属性：攻击力，生命值和速度。一张卡牌能胜过另一张卡牌，当且仅当这张卡牌在至少两个属性上要严格大于另一张卡牌。

通过某些不可告人的手段，小 D 已经知道了小 B 有 N 张卡牌，第 i 张卡牌的攻击力、生命值和速度分别为： a_i, b_i, c_i 。小 D 想要找出一张卡牌，使得它能胜过所有小 B 的卡牌。

小 D 一共拥有 $A \times B \times C$ 张卡牌，即攻击力从 1 到 A 、生命值从 1 到 B 、速度从 1 到 C 的卡牌各有一张。他想要知道，有多少张他的卡牌能胜过小 B 所有的卡牌。

2.2 输入格式

从文件 card.in 中读取数据。

第一行四个整数 N, A, B, C ，表示小 B 卡牌的数量，小 D 拥有的卡牌的攻击力、生命值、速度的最大值。

第二行到第 $N + 1$ 行，每行三个整数 a_i, b_i, c_i ，描述小 B 的一张卡牌。

2.3 输出格式

输出到文件 card.out 中。

一行一个整数，描述小 D 能胜过小 B 所有卡牌的卡牌数量。

2.4 样例输入 1

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

2.5 样例输出 1

```
10
```

2.6 样例解释

以下 10 张卡牌可以胜过小 B 的所有卡牌： $(3, 3, 5), (3, 4, 2), (3, 4, 3), (3, 4, 4), (3, 4, 5), (4, 3, 5), (4, 4, 2), (4, 4, 3), (4, 4, 4), (4, 4, 5)$ 。

2.7 样例输入 2

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

2.8 样例输出 2

```
152
```

2.9 数据范围与约定

对于前 20% 的数据， $N \leq 20, A, B, C \leq 100$ ；

对于前 40% 的数据， $N \leq 2000$ ；

对于另外 20% 的数据， $N \leq 10^5$ ，所有的 c_i 都相同；

对于前 80% 的数据， $N, A, B, C \leq 10^5$ ；

对于 100% 的数据， $N, A, B, C \leq 7 \times 10^5, 1 \leq a_i, b_i, c_i \leq 7 \times 10^5$ ，满足 $a_i \leq A, b_i \leq B, c_i \leq C$ 。

3 方形的零 (zero)

3.1 题目描述

小 D 有一个 $2 \times N$ 的矩阵，矩阵中每个元素是正数、负数或者零。小 D 想要选出尽可能多的矩形，使得每个矩形的和为 0 且矩形间没有公共元素。

小 D 想要知道在这个矩阵中最多可以选出多少个矩形？

3.2 输入格式

从文件 zero.in 中读取数据。

第一行一个整数 N ，表示矩阵的长度。

第二行 N 个正整数 a_{11}, \dots, a_{1N} ，描述矩阵的第一行；

第三行 N 个正整数 a_{21}, \dots, a_{2N} ，描述矩阵的第二行。

3.3 输出格式

输出到文件 zero.out 中。

一行一个整数，描述最多能选出多少个矩形。

3.4 样例输入 1

```
6
70 70 70 70 70 -15
90 -60 -30 30 -30 15
```

3.5 样例输出 1

```
3
```

3.6 样例解释

一共有 4 个矩形的和为 0：

70	70	70	70	70	-15
90	-60	-30	30	-30	15

70	70	70	70	70	-15
90	-60	-30	30	-30	15

70	70	70	70	70	-15
90	-60	-30	30	-30	15

70	70	70	70	70	-15
90	-60	-30	30	-30	15

由于矩形间有交点，不能同时选择这四个矩形。可以同时选择画蓝框的三个矩形。

3.7 样例输入 2

```
4
0 -1 0 0
0 0 1 0
```

3.8 样例输出 2

```
6
```

3.9 数据范围与约定

对于前 10% 的数据， $N \leq 10$ ；

对于前 30% 的数据， $N \leq 10^3$ ；

对于另外 15% 的数据， $|a_i| \leq 1$ ；

对于另外 15% 的数据，保证在同一行中连续 10 个数中一定存在一个 0；

对于 100% 的数据， $N \leq 10^6$ ， $-10^9 \leq a_i \leq 10^9$ 。