



2024 Loi Summer Final Round

题目名称	数组	网格	普通平衡树	CS:GO	一方通行
可执行文件名	array	grid	balance	fight	accelerator
输入输出文件名	array.in/ out	grid.in/ out	balance.in/ out	fight.in/ out	accelerator.in/ out
每个测试点时限	1s	1s	1s	1s	1s
每个测试点内存	256MB	256MB	256MB	256MB	256MB
题目分值	40	60	100	100	100

温馨提示

推荐使用万能头文件：`#include<bits/stdc++.h>`

文件输入输出

```
freopen("xxx.in", "r", stdin);  
freopen("xxx.out", "w", stdout);
```

其中的 `xxx` 为文件名。这两行需放在 `main()` 函数的开头，请确保读入之前已进入开启文件操作的状态。

以 **A+B** 为例（文件名为 `flashbang`）

```
#include<bits/stdc++.h>  
using namespace std;  
  
int main()  
{  
    freopen("flashbang.in", "r", stdin);  
    freopen("flashbang.out", "w", stdout);  
    int a,b;  
    cin>>a>>b;  
    cout<<a+b;  
    fclose(stdin);  
    fclose(stdout);  
    return 0;  
}
```

数组 (array)

题目描述

Qdcc 正在模仿御坂美琴弹硬币。

他有一个长度为 n 的数组 a ，但他并不满足，他会把一个标有数字 x 的硬币弹到这个数组的第 k 个元素之后。形式化地说，请帮他在 a 的第 k 个元素后面插入一个 x 并输出。

输入格式

数据将以以下形式给出：

$N K X$

$A_1 A_2 \dots A_N$

输出格式

请输出修改过后的得到的数组 B ，元素之间以空格隔开

$B_1 B_2 \dots B_{N+1}$

样例 #1

样例输入 #1

```
4 3 7
2 3 5 11
```

样例输出 #1

```
2 3 5 7 11
```

样例 #2

样例输入 #2

```
1 1 100
100
```

样例输出 #2

```
100 100
```

样例 #3

样例输入 #3

```
8 8 3
9 9 8 2 4 4 3 5
```

样例输出 #3

```
9 9 8 2 4 4 3 5 3
```

提示

数据范围

输入的数保证全为整数

$$1 \leq K \leq N \leq 100$$

$$1 \leq A_i, X \leq 100$$

网格 (grid)

题目描述

LightFive 有一个由 N 行 M 列的空白正方形组成的网格。

他懒得动弹，但是他想把一部分正方形涂成黑色，于是需要你的帮助。你可以进行以下两种操作：

- 选择一行，然后将该行中的所有正方形涂成黑色。
- 选择一列，然后将该列中的所有正方形涂成黑色。

为了在网格中具有 K 个或更多的黑色正方形，你至少需要进行多少次操作？

LightFive 保证，他的网格上的正方形数量不多于 K 。

输入格式

输入一行由空格隔开的三个数。

```
 $M N K$ 
```

输出格式

输出操作次数的最小值。

样例 #1

样例输入 #1

```
3  
7  
10
```

样例输出 #1

```
2
```

样例 #2

样例输入 #2

```
14  
12  
112
```

样例输出 #2

```
8
```

样例 #3

样例输入 #3

```
2  
100  
200
```

样例输出 #3

```
2
```

提示

对于 10% 的数据, 保证 $N = M = K = 1$;

对于另外 20% 的数据, 保证 $K = M \times N$;

对于 100% 的数据保证 $N, M \leq 1000$ 。

普通平衡树 (balance)

题目描述

wyz 想写平衡树，但是他不会写；于是 Ye_Zhui_Yi 提出了一个解决方案：由你来完成平衡树的操作，把你变成一颗平衡树就能完美解决问题！

而一颗平衡树可实现的操作如下：

1. 插入一个数 x 。
2. 删除一个数 x （若有多个相同的数，应只删除一个）。
3. 定义排名为比当前数小的数的个数 +1。查询 x 的排名。
4. 查询平衡树中排名为 x 的数。
5. 求 x 的前驱（前驱定义为小于 x ，且最大的数）。
6. 求 x 的后继（后继定义为大于 x ，且最小的数）。

对于操作 3,5,6, **不保证**当前数据结构中存在数 x 。

请注意数据范围

输入格式

第一行为 n ，表示操作的个数，下面 n 行每行有两个数 opt 和 x ， opt 表示操作的序号 ($1 \leq opt \leq 6$)。

输出格式

对于操作 3, 4, 5, 6 每行输出一个数，表示对应答案。

样例 #1

样例输入 #1

```
10
1 106465
4 1
1 317721
1 460929
1 644985
1 84185
1 89851
6 81968
1 492737
5 493598
```

样例输出 #1

```
106465
84185
492737
```

数据范围

对于 5% 的数据，不存在 3, 4, 5, 6 操作；

对于另外 20% 的数据，不存在 3, 4 操作；

对于另外 20% 的数据，不存在 5, 6 操作；

对于另外 25% 的数据， $n \leq 100$ ；

对于 100% 的数据， $1 \leq n \leq 10^3$ ， $|x| \leq 10^7$ 。

CS:GO (fight)

题目描述

Fox_konata 和 Hywel 正在玩一款叫 Counter – Strike : Global Offensive 的游戏, 游戏的棋盘是一条线段, 线段上有 n 个兵营 (自左至右编号 $1 \sim n$), 相邻编号的兵营之间相隔 1 厘米, 即棋盘为长度为 $n - 1$ 厘米的线段。 i 号兵营里有 c_i 位探员。下面图 1 为 $n = 6$ 的示例:

alt text

Fox_konata 在左侧, 代表“CT”; Hywel 在右侧, 代表“T”。他们以 m 号兵营作为分界, 靠左的探员属于CT势力, 靠右的探员属于T势力, 而第 m 号兵营中的探员很纠结, 他们不属于任何一方。

一个兵营的气势为: 该兵营中的探员数 \times 该兵营到 m 号兵营的距离; 参与游戏一方的势力定义为: 属于这一方所有兵营的气势之和。

下面图 2 为 $n = 6, m = 4$ 的示例, 其中红色为CT方, 黄色为T方:

alt text

游戏过程中, 某一刻天降神兵, 共有 s_1 位探员突然出现在了 p_1 号兵营。作为 Fox_konata 和 Hywel 的朋友, 你知道如果CT与T双方气势差距太悬殊, Fox_konata 和 Hywel 就不愿意继续玩下去了。为了让游戏继续, 你需要选择一个兵营 p_2 , 并将你手里的 s_2 位探员全部派往兵营 p_2 , 使得双方气势差距尽可能小。

注意: 你手中的探员落在哪个兵营, 就和该兵营中其他探员有相同的势力归属 (如果落在 m 号兵营, 则不属于任何势力)。

输入格式

输入文件的第一行包含一个正整数 n , 代表兵营的数量。

接下来的一行包含 n 个正整数, 相邻两数之间以一个空格分隔, 第 i 个正整数代表编号为 i 的兵营中起始时的探员数量 c_i 。

接下来的一行包含四个正整数，相邻两数间以一个空格分隔，分别代表 m, p_1, s_1, s_2 。

输出格式

输出文件有一行，包含一个正整数，即 p_2 ，表示你选择的兵营编号。如果存在多个编号同时满足最优，取最小的编号。

样例 #1

样例输入 #1

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

样例输出 #1

```
2
```

样例 #2

样例输入 #2

```
6
1 1 1 1 1 16
5 4 1 1
```

样例输出 #2

```
1
```

提示

样例 1 说明

见问题描述中的图 2。

双方以 $m = 4$ 号兵营分界，有 $s_1 = 5$ 位探员突然出现在 $p_1 = 6$ 号兵营。

CT方的气势为：

$$2 \times (4 - 1) + 3 \times (4 - 2) + 2 \times (4 - 3) = 14$$

T方的气势为：

$$2 \times (5 - 4) + (3 + 5) \times (6 - 4) = 18$$

当你将手中的 $s_2 = 2$ 位探员派往 $p_2 = 2$ 号兵营时，CT方的气势变为：

$$14 + 2 \times (4 - 2) = 18$$

此时双方气势相等。

样例 2 说明

双方以 $m = 5$ 号兵营分界，有 $s_1 = 1$ 位探员突然出现在 $p_1 = 4$ 号兵营。

CT方的气势为：

$$1 \times (5 - 1) + 1 \times (5 - 2) + 1 \times (5 - 3) + (1 + 1) \times (5 - 4) = 11$$

T方的气势为：

$$16 \times (6 - 5) = 16$$

当你将手中的 $s_2 = 1$ 位探员派往 $p_2 = 1$ 号兵营时，CT方的气势变为：

$$11 + 1 \times (5 - 1) = 15$$

此时可以使双方气势的差距最小。

数据规模与约定

$$1 < m < n, 1 \leq p_1 \leq n。$$

对于 20% 的数据, $n = 3, m = 2, c_i = 1, s_1, s_2 \leq 100。$

另有 20% 的数据, $n \leq 10, p_1 = m, c_i = 1, s_1, s_2 \leq 100。$

对于 60% 的数据, $n \leq 100, c_i = 1, s_1, s_2 \leq 100。$

对于 80% 的数据, $n \leq 100, c_i, s_1, s_2 \leq 100。$

对于 100% 的数据, $n \leq 10^5, c_i, s_1, s_2 \leq 10^9。$

一方通行 (accelerator)

题目背景

日语中的“一方通行”也有单行道的意思。

题目描述

由于今年是《魔法禁书目录》出版二十周年，学园都市内的 n 个车站正在举办庆祝活动，为每个过往的行人发纪念徽章。每个车站只有 1 种纪念徽章，而这些纪念徽章共有 k 种。

很可惜，连接这 n 个车站的 m 条线路都是单向通行。为了防止部分人多次到访同一个站点，管理人员在设置线路的时候就保证不会出现环。

Zoomy 现在在 1 号站点，可以任意选择终点。请你计算 Zoomy 最多拿到多少个同种徽章。

输入格式

第一行三个数 n, m, k 。

第二行 n 个数 a_i ，表示第 i 个站点拥有的徽章类型。

之后的 m 行每行两个数 u, v ，代表一条从 u 到 v 的**单行道**。

输出格式

一个数，表示同种徽章数量的最大值。

样例 #1

样例输入 #1

```
5 7 2
1 0 0 1 1
1 2
1 4
4 2
2 3
4 3
3 5
1 5
```

样例输出 #1

```
3
```

样例 #2

样例输入 #2

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

样例输出 #2

```
2
```

请注意，该组样例满足 SubTask 2 的条件。

样例 #3

样例输入 #3

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

样例输出 #3

```
1
```

请注意，该组样例满足 SubTask 1 的条件。

数据范围

对于 5% 的数据， $n = 2$ ；

对于 20% 的数据， $n \leq 10$ ；

对于另外 10% 的数据， $m = n - 1$ 且第 i 条边连接 1 与 i ；(SubTask 1 捆绑测试)

对于另外 30% 的数据， $m = n - 1$ 且第 i 条边连接 i 与 $i + 1$ ；(SubTask 2 捆绑测试)

对于全部的数据， $n \leq 10^5$ ， $m \leq \min\{\frac{n \times (n - 1)}{2}, 5 \times 10^5\}$ ， $k \leq 10$ 。