

# NOIP模拟赛

Wuweizheng

题目名称	a	b	c
英文名称	a	b	c
源程序名	a.cpp	b.cpp	c.cpp
输入文件	a.in	b.in	c.in
输出文件	a.out	b.out	c.out
时间限制	2s	2s	1s
内存限制	512MB	512MB	512MB
比较方式	全文比较	全文比较	全文比较
编译开关	-O2 -lm -std=c++11	-O2 -lm -std=c++11	-O2 -lm -std=c++11
题目类型	传统	传统	传统

## 1.a

---

### 1.1 Description

小 A 同学得到了一条长为  $L$  米的白色绳子，但是他觉得这条绳子的颜色过于单调了，于是他决定给绳子染上色让它变成一条彩绳。小 A 一共有  $m$  种颜色，他将  $L$  米长的绳子平均划分成了  $L$  小段，显然，每一小段的长度为 1 米，然后给每一小段都染上一种颜色，每种颜色都可以染无数次，不必全部使用。

但是考虑到马上就要举办同学聚会了，于是他决定将这条长为  $L$  米的绳子顺次截成  $n$  条，第  $i$  条的长度为  $l_i$  米，用来送给  $n$  个好朋友。为了让他的好朋友对他们收到的礼物很满意，小 A 觉得，送给每个人的单独的一条绳子中，不能出现相邻的两小段颜色相同的情况。同时，为了让同学们感觉到他收到的礼物很特殊，小 A 想让任意相邻的两条绳子的颜色种类不完全相同，相邻两条绳子颜色种类不完全相同指的是至少存在一种颜色，使得它不会同时出现在相邻两条绳子上。

那么，现在小 A 想知道，他有多少种不同的染色方案，两种染色方案不同，指的是原来在初始的  $L$  中的位置上至少存在一个小段使得它所染的颜色不相同。

### 1.2 Input Format

输入共两行：

第一行三个正整数  $n, m, p$  分别表示分成的绳子的条数，颜色的种类数，以及模数。

第二行  $n$  个正整数  $l_i$ ，表示截成的每一条绳子的长度。

### 1.3 Output Format

输出共一行：

一个整数，表示不同的染色方案数，答案对  $p$  取模。

### 1.4 Sample 1

#### 1.4.1 Input

```
3 2 1000
3 1 2
```

### 1.4.2 Output

```
8
```

### 1.4.3 Explanation

如下是8种染色方法 :(1, 2 表示两种颜色)

(121,1,12),(121,1,21),(121,2,12),(121,2,21),(212,1,12),(212,1,21),(212,2,12),(212,2,21)

## 1.5 Sample2

### 1.5.1 Input

```
5 3 100
1 1 1 1 1
```

### 1.5.2 Output

```
48
```

## 1.6 Sample3

见选手目录下的 *a/a3.in* 与 *a/a3.out*.

这个样例，无疑是善良出题人无私的馈赠。精心构造的  $n = 10^6, m = 10^6$  的极限测试数据，涵盖了测试点中所有出现性质的组合，你可以利用这个测试点，对自己的程序进行全面的检查。足量的数据组数、不大的数据范围和多种多样的数据类型，能让程序的错误无处遁形。出题人相信，这个美妙的样例，可以给拼搏于 AC 这道题的逐梦之路上的你提供一个用力的援助！

## 1.7 Constraints

本题共20个测试点。

- 对于 30% 的数据：  $n, m \leq 5, L \leq 10$
- 对于 30% 的数据：  $n = 1, m \leq 10^6, 2 \leq p \leq 10^9, 1 \leq l_i \leq 5000, L = \sum_{i=1}^n l_i \leq 10^7$
- 对于 10% 的数据：  $1 \leq n, m \leq 10^6, 2 \leq p \leq 10^9, 1 \leq l_i \leq 5, L = \sum_{i=1}^n l_i \leq 10^7$
- 对于 30% 的数据：  $1 \leq n, m \leq 10^6, 2 \leq p \leq 10^9, 1 \leq l_i \leq 5000, L = \sum_{i=1}^n l_i \leq 10^7$

## 2.b

---

### 2.1 Description

给出一颗  $n$  个节点的以 1 为根的有根树，树的节点从 1 到  $n$  编号，每个节点的初始权值即为节点的编号，同时每个节点有一个  $ans$  值，初始时均为 0。树上有一些关键点。有  $m$  个操作，每个操作有如下两种：

- 1  $x$  对于树中的每一个关键点，在它与节点  $x$  的  $LCA$  的  $ans$  值上加上该关键点的权值。
- 2  $x$  修改节点  $x$  的状态。如果节点  $x$  原来是一个关键点，则改为不是关键点；如果节点  $x$  原来不是一个关键点，则让它成为关键点。

现在，请你求出在执行完所有的  $m$  个操作之后，求每个节点的  $ans$  值。

### 2.2 Input Format

输入共  $n + m + 1$  行：

第一行两个整数  $n, m$ ，分别表示树的节点数，以及操作的个数。

第二行  $n$  个整数  $a_1, a_2, \dots, a_n$ ，对于  $\forall a_i \in [0, 1]$ 。如果  $a_i$  为 0，则表示节点  $i$  不是关键点；如果  $a_i$  为 1，则表示节点  $i$  是关键点。

第 3... $n + 1$  行，共  $n - 1$  行，每行两个整数  $x, y$ ，表示节点  $x$  与节点  $y$  有边相连。

第  $n + 2$ ... $n + m + 1$  行，共  $m$  行，每行一个操作，形式如题目描述所示。

### 2.3 Output Format

输出共  $n$  行：

每行一个整数，第  $i$  行表示执行完所有的  $m$  个操作之后节点  $i$  的  $ans$  值。

### 2.4 Sample1

#### 2.4.1 Input

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

## 2.4.2 Output

```
4
0
13
0
0
```

## 2.4.3 Explanation

开始的时候，只有节点 2 和节点 4 为关键点，每个节点的 *ans* 值均为 0。

执行操作 1 5 之后，即对节点 2 与节点 5 的 *LCA* 节点 1 的 *ans* 加上 2，对节点 4 与节点 5 的 *LCA* 节点 3 的 *ans* 加上 4。所以每个节点的 *ans* 值分别为 2 0 4 0 0。

执行操作 2 5 之后，节点 5 变为关键点，则有节点 2, 4, 5 为关键点。

执行操作 1 3 之后，即对节点 2 与节点 3 的 *LCA* 节点 1 的 *ans* 加上 2，对节点 4 与节点 3 的 *LCA* 节点 3 的 *ans* 加上 4，对节点 5 与节点 3 的 *LCA* 节点 3 的 *ans* 加上 5。所以每个节点的 *ans* 值分别为 4 0 13 0 0。

## 2.5 Sample 2

### 2.5.1 Input

```
15 8
0 1 0 0 0 0 0 1 0 0 0 1 0 0 1
1 2
1 3

1 4
```

```
2 5
2 6
3 7
7 8
7 9
7 10
9 11
9 12
4 13
13 14
13 15
1 11
1 5
2 13
2 14
1 14
2 9
2 15
1 6
```

## 2.5.2 Output

```
130
4
0
0
0
0
8
0
12
0
0
0
0
28
14
0
```

## 2.6 Sample 3

见选手目录下的 *b/b3.in* 与 *b/b3.out*

这个样例，无疑是善良出题人无私的馈赠。精心构造的  $n = 3 * 10^5, m = 3 * 10^5$  树的形态随机的极限测试数据，涵盖了测试点中所有出现性质的组合，你可以利用这个测试点，对自己的程序进行全面的检查。足量的数据组数、不大的数据范围和多种多样的数据类型，能让程序的错误无处遁形。出题人相信，这个美妙的样例，可以给拼搏于AC这道题的逐梦之路上的你提供一个用力的援助！

## 2.7 Constraints

本题共10个测试点

- 对于30%的数据： $1 \leq n, m \leq 3000$
- 对于30%的数据： $1 \leq n, m \leq 300000$ ，树的形态随机。
- 对于40%的数据： $1 \leq n, m \leq 300000$



## 3.c

### 3.1 Description

小  $B$  有一个  $n \times 4$  的  $n$  行 4 列的环形棋盘，每个格子中都有一个  $1 \dots 4 \times n$  的数，每个格子中只有一个数，且每个数只会出现一次。定义一个棋盘的初始状态为，将  $1 \dots 4 \times n$  个数从左到右，从上到下地顺次放入棋盘中的状态。例如， $n = 4$  时的初始状态为：

```
1  2  3  4
5  6  7  8
9 10 11 12
13 14 15 16
```

但是由于小  $A$  同学很调皮，将初始状态打乱了，现在小  $B$  想将棋盘移动回初始状态，但是小  $A$  肯定不会让他这么轻易地达到目的，所以他要求小  $B$  只能按照他的要求移动：每次只能将数 1 向上下左右四个方向中的一个移动一格，并将原来在那一格的数放到 1 的位置上。即如果 1 当前在  $(x, y)$ ，那么它下一步可以跳到  $(x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1)$  中的一个；另外，由于棋盘是环形的，所以如果它在第 1 列，那么它向左移动一格之后，就会到第 4 列，如果它在第 4 列，那么它向右移动一格之后，就会到第 1 列。但是要注意，上下是不会连接的，即如果 1 当前在第 1 行，则不能将它向上移动了，同时，如果在第  $n$  行，也不能再向下移动了。

现在小  $B$  想知道，他需要多少步才能将棋盘移动回初始状态。

### 3.2 Input Format

输入共  $n + 1$  行：

第一行一个整数  $n$  表示棋盘的行数。

第二行到第  $n + 1$  行，共  $n$  行，每行 4 个整数，描述这个被小  $A$  打乱之后的环形棋盘。

注意：输入数据保证有解。

### 3.3 Output Format

一行一个整数。输出最小的移动步数。

## 3.4 Sample1

### 3.4.1 Input

```
1
1 2 3 4
```

### 3.4.2 Output

```
0
```

### 3.4.3 Explanation

这显然不需要移动。

## 3.5 Sample2

### 3.5.1 Input

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

### 3.5.2 Output

```
2
```

### 3.5.3 Explanation

先将 1 向右移动一格，与 8 交换，再向上移动一格，与 5 交换。一共 2 步，即可回到初始状态。

### 3.6 Sample3

见选手目录下  $c/c3.in - c/c10.in$  与  $c/c3.out - c/c10.out$

这些样例，无疑是善良出题人无私的馈赠。精心构造的  $n = 1 - 10$  的测试数据，涵盖了测试点中所有出现性质的组合，你可以利用这些测试点，对自己的程序进行全面的检查。足量的数据组数、不大的数据范围和多种多样的数据类型，能让程序的错误无处遁形。出题人相信，这些美妙的样例，可以给拼搏于 AC 这道题的逐梦之路上的你提供一个用力的援助！

### 3.7 Constraints

为了防止暴力水过，故善良的出题人决定采用捆绑测试。

- $Subtask1(10pts) : n \leq 1$
- $Subtask2(10pts) : n \leq 2$
- $Subtask3(10pts) : n \leq 3$
- $Subtask4(10pts) : n \leq 4$
- $Subtask5(10pts) : n \leq 5$
- $Subtask6(10pts) : n \leq 6$
- $Subtask7(10pts) : n \leq 7$
- $Subtask8(10pts) : n \leq 8$
- $Subtask9(10pts) : n \leq 9$
- $Subtask10(10pts) : n \leq 10$