



VJudge 天涯未找到

M+

## B. Siga ta Kymata

time limit per test: 1.5 seconds

memory limit per test: 256 megabytes

M-

You are given a permutation\*  $p$  of every integer from 1 to  $n$ . You also own a binary† string  $s$  of size  $n$  where  $s_i = 0$  for all  $1 \leq i \leq n$ . You may do the following operation at most 5 times:

- Choose any two integers  $l$  and  $r$  such that  $1 \leq l \leq r \leq n$ . Then, for every  $i$  such that  $l < i < r$  and  $\min(p_l, p_r) < p_i < \max(p_l, p_r)$  hold at the same time, you will set  $s_i$  to 1.

You are also given a binary string  $x$  of size  $n$ . After performing operations, it must hold for every  $1 \leq i \leq n$  that if  $x_i = 1$ , then  $s_i = 1$ . Note that if  $x_i = 0$ , then  $s_i$  can have **any** value.

Figure out any sequence of **at most 5** operations such that the aforementioned condition is satisfied, or report that it is impossible to do so. Note that you **do not** have to minimize the number of operations you make.

\*A permutation  $p$  of every integer from 1 to  $n$  is a sequence of elements from 1 to  $n$  such that every element appears exactly once.

†A string  $b$  of size  $m$  is considered binary if and only if  $b_i = 0$  or  $b_i = 1$  for all  $1 \leq i \leq m$ .

有道翻译

□ ×

您将得到从 1 到  $n$  的每个整数的一个排列 \*  $p$ 。您还拥有一个大小为  $n$  的二进制字符串 †  $s$ ，其中  $s_i = 0$  表示所有  $1 \leq i \leq n$ 。您最多可以执行 5 次以下操作：

### Codeforces Round 1063 (Div. 2)

比赛进行中

01:46:42

Contestant



### → 提交?

语言:  GNU G++20 13.2 (64 bit, ✓)

选择文件:  未选择文件

Be careful: there is 50 points penalty for submission which fails the pretests or resubmission (except failure on the first test, denial of judgement or similar verdicts). "Passed pretests" submission verdict doesn't guarantee that the solution is absolutely correct and it will pass system tests.

提交

### → 评分表

Score

[Problem A](#)

488



—选择  $1 \leq l \leq r \leq n$  的任意两个整数  $l$  和  $r$ 。然后，对于  $l < i < r$  和  $\min(p_l, p_r) < p_i < \max(p_l, p_r)$  同时存在的  $i$ ，将  $s_i$  设置为 1。

您还会得到一个大小为  $n$  的二进制字符串  $x$ 。在执行操作后，它必须为每个  $1 \leq i \leq n$  保持  $x_i = 1$ ，然后  $s_i = 1$ 。注意，如果  $x_i = 0$ ，那么  $s_i$  可以有任意值。

求出的任意一个最多满足 5 操作的序列，或者报告不可能满足上述条件。请注意，您不必减少所执行的操作数量。

\* 从 1 到  $n$  的每个整数的  $p$  的排列是从 1 到  $n$  的元素序列，使得每个元素只出现一次。

† 大小为  $m$  的字符串  $b$  被认为是二进制的，当且仅当  $b_i = 0$  或  $b_i = 1$  对于所有  $1 \leq i \leq m$ 。



\* If you solve problem on 00:06 from the first attempt

## Input

Each test contains multiple test cases. The first line contains the number of test cases  $t$  ( $1 \leq t \leq 10^4$ ). The description of the test cases follows.

The first line of each test case contains a single integer  $n$  ( $3 \leq n \leq 2 \cdot 10^5$ ) — the size of the array.

The second line contains exactly  $n$  integers  $p_1, p_2, \dots, p_n$  ( $1 \leq p_i \leq n$ , the elements of  $p$  are pairwise distinct) — where  $p_i$  is the  $i$ -th element of the permutation.

The third line contains a single binary string  $x$  of size  $n$ .

It is guaranteed that the sum of  $n$  over all test cases does not exceed  $2 \cdot 10^5$ .

有道 翻译 □ ×

输入 \*\*\*

每个测试包含多个测试用例。第一行包含测试用例的数量  $t$  ( $1 \leq t \leq 10^4$ )。下面是测试用例的描述。

每个测试用例的第一行包含一个整数  $n$  ( $3 \leq n \leq 2 \cdot 10^5$ ) ——数组的大小。

第二行恰好包含  $n$  个整数  $p_1, p_2, \dots, p_n$  ( $1 \leq p_i \leq n$ ， $p$  的元素是成对不同的) —其中  $p_i$  是排列的第  $i$  个元素。

第三行包含一个大小为  $n$  的二进制字符串  $x$ 。

<a href="#">Problem B</a>	976
<a href="#">Problem C</a>	1220
<a href="#">Problem D1</a>	1464
<a href="#">Problem D2</a>	1220
<a href="#">Problem E</a>	2684
Successful hack	100
Unsuccessful hack	-50
Unsuccessful submission	-50
Resubmission	-50



保证所有测试用例  $n$  的和不超过  $2 \cdot 10^5$ 。



## Output

For each test case, if it is impossible to perform operations such that the constraint is satisfied, output  $-1$ .

Otherwise, output an integer  $0 \leq k \leq 5$ , the number of operations. On the  $i$ -th of the next  $k$  lines, output two integers  $1 \leq l_i \leq r_i \leq n$ , the bounds of the  $i$ -th operation that is performed. If there are multiple correct solutions, output any of them.

有道 翻译



\*\* \*\*输出

对于每个测试用例，如果不可能执行满足约束的操作，则输出  $-1$ 。

否则，输出一个整数  $0 \leq k \leq 5$ ，表示操作次数。在接下来的  $k$  行的第  $i$  上，输出两个整数  $1 \leq l_i \leq r_i \leq n$ ，这是所执行的第  $i$  次操作的边界。如果有多个正确的解决方案，输出其中的任何一个。

## Example

input

Copy

```
6
3
1 2 3
010
5
3 4 2 1 5
11111
6
1 3 2 4 6 5
001100
6
6 2 3 4 5 1
110110
5
2 1 4 3 5
00000
5
2 5 3 1 4
00100
```



## output

Copy

```
1  
1 3  
-1  
2  
1 5  
2 6  
-1  
0  
1  
2 4
```



### Note

In the first example,  $p = [1, 2, 3]$ , and  $x = 010$ . We can perform a single operation, with  $l = 1$  and  $r = 3$ . After the operation, we set  $s_2$  to 1 since  $l < 2 < r$  and  $\min(p_l, p_r) < p_2 = 2 < \max(p_l, p_r)$  hold at the same time. As a result,  $s = 010$ .

In the second example, it can be shown that there does not exist a correct sequence of at most 5 operations, so we output  $-1$ .

In the third example,  $p = [1, 3, 2, 4, 6, 5]$  and  $x = 001100$ . After performing an operation for  $l = 1$  and  $r = 5$ , then  $s = 011100$ . If we also perform an operation for  $l = 2$  and  $r = 6$ , then  $s$  will remain the same. The string  $s = 011100$  is valid, because for every position where  $x$  has a 1, then  $s$  also has a 1 at that position.

有道翻译



### 注意

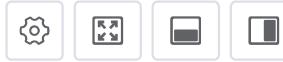
在第一个例子中， $p = [1, 2, 3]$  和  $x = 010$ 。我们可以使用  $l = 1$  和  $r = 3$  执行单个操作。操作完成后，我们将  $s_2$  设置为 1，因为  $l < 2 < r$  和  $\min(p_l, p_r) < p_2 = 2 < \max(p_l, p_r)$  同时保持。因此， $s = 010$ 。

在第二个示例中，可以看出不存在最多 5 个操作的正确序列，因此我们输出  $-1$ 。

在第三个例子中， $p = [1, 3, 2, 4, 6, 5]$  和  $x = 001100$ 。在对  $l = 1$  和  $r = 5$  执行操作后，再执行  $s = 011100$ 。如果我们也对  $l = 2$  和  $r = 6$  执行操作，那么  $s$  将保持不变。字符串  $s = 011100$  是有效的，因为对于每个位置  $x$  有一个 1，那么在该位置  $s$  也有一个 1。



GNU G++20 13.2 (64 bit, winlibs)



1

▶ 自定义测试数据(自动保存)

▶✉

[Codeforces](#) (c) Copyright 2010-2025 Mike Mirzayanov

The only programming contests Web 2.0 platform

Server time: Nov/10/2025 23:11:04<sup>UTC+8</sup> (I3).

Desktop version, switch to [mobile version](#).

[Privacy Policy](#) | [Terms and Conditions](#)

Supported by



| **ITMO**