

## E. Lost Soul

time limit per test: 2 seconds

memory limit per test: 256 megabytes

You are given two integer arrays  $a$  and  $b$ , each of length  $n$ .

You may perform the following operation any number of times:

- Choose an index  $i$  ( $1 \leq i \leq n - 1$ ), and set  $a_i := b_{i+1}$ , or set  $b_i := a_{i+1}$ .

Before performing any operations, you are allowed to choose an index  $i$  ( $1 \leq i \leq n$ ) and remove both  $a_i$  and  $b_i$  from the arrays. This removal can be done at most once.

Let the number of matches between two arrays  $c$  and  $d$  of length  $m$  be the number of positions  $j$  ( $1 \leq j \leq m$ ) such that  $c_j = d_j$ .

Your task is to compute the maximum number of matches you can achieve.

DeepL 翻译

给你两个长度分别为  $n$  的整数数组  $a$  和  $b$ 。

您可以多次执行以下操作：

- 选择索引  $i$  ( $1 \leq i \leq n - 1$ )，并设置  $a_i := b_{i+1}$ ，或设置  $b_i := a_{i+1}$ 。

在执行任何操作之前，允许选择索引  $i$  ( $1 \leq i \leq n$ ) 和设置  $b_i := a_{i+1}$ 。( $1 \leq i \leq n$ ) 并同时删除数组中的  $a_i$  和  $b_i$ 。这种删除操作最多只能进行\*\*次。

假设长度为  $m$  的两个数组  $c$  和  $d$  之间的匹配数是位置  $j$  ( $1 \leq j \leq m$ ) 的个数。 $(1 \leq j \leq m)$  这样的位置数  $c_j = d_j$ 。

你的任务是计算所能达到的最大匹配数。

### Input

The first line of the input contains an integer  $t$  ( $1 \leq t \leq 10^4$ ) — the number of test cases. The description of each test case follows.

The first line contains an integer  $n$  ( $2 \leq n \leq 2 \cdot 10^5$ ) — the length of  $a$  and  $b$ .

The second line contains  $n$  integers  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq n$ ) — the elements of  $a$ .

The third line contains  $n$  integers  $b_1, b_2, \dots, b_n$  ( $1 \leq b_i \leq n$ ) — the elements of  $b$ .

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

DeepL 翻译

### 输入

输入的第一行包含一个整数  $t$  ( $1 \leq t \leq 10^4$ ) - 测试用例的数量。( $1 \leq t \leq 10^4$ ) - 测试用例的数量。每个测试用例的说明如下。

第一行包含一个整数  $n$  ( $2 \leq n \leq 2 \cdot 10^5$ ) -  $n$  的长度。( $2 \leq n \leq 2 \cdot 10^5$ ) -  $a$  和  $b$  的长度。

第二行包含  $n$  个整数  $a_1, a_2, \dots, a_n$  - ( $1 \leq a_i \leq n$ ) 的长度。( $1 \leq a_i \leq n$ ) -  $a$  的元素。

### Codeforces Round 1029 (Div. 3)

Finished

Practice



### → 什么是虚拟参赛

虚拟比赛是一种参加过去比赛的方式，尽可能接近实时参赛。目前只支持ICPC模式的虚拟赛事。如果你已经看过这些题目，虚拟赛事就不适合你——应该在题库中解决这些题目。如果你只是想解决比赛中的某个问题，虚拟赛事也不适合你——应该在题库中解决这个问题。在虚拟比赛期间，切勿使用别人的代码，阅读题解或与他人交流。

[开始虚拟参赛](#)

### → 克隆比赛到组合混搭

您可以将此比赛克隆到混搭比赛中。

[克隆比赛](#)

### → 提交?

语言: [GNU G++17 7.3.0](#)

选择文件:  未选择文件

[提交](#)

### → 问题标签

[暴力枚举 \(brute force\)](#) [贪心 \(greedy\)](#)

\*1600

没有标签编辑权限

### → 比赛相关资料

- 公告 (en) 
- 题解 (en) 

第三行包含  $n$  个整数  $b_1, b_2, \dots, b_n$  - ( $1 \leq b_i \leq n$ ) 的元素。( $1 \leq b_i \leq n$ ) -  $b$  的元素。

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



## Output

For each test case, print a single integer — the answer for the test case.

DeepL 翻译



## 输出

为每个测试用例打印一个整数，即测试用例的答案。

## Example

### input

Copy

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

### output

Copy

```
3
3
0
4
3
5
6
4
5
2
```



## Note

In the first test case, we can do the following:

- We will choose not to remove any index.
- Choose index 3, and set  $a_3 := b_4$ . The arrays become:  $a = [1, 3, 2, 4]$ ,  $b = [4, 3, 2, 2]$ .
- Choose index 1, and set  $a_1 := b_2$ . The arrays become:  $a = [3, 3, 2, 4]$ ,  $b = [4, 3, 2, 2]$ .
- Choose index 1, and set  $b_1 := a_2$ . The arrays become:  $a = [3, 3, 2, 4]$ ,  $b = [3, 3, 2, 2]$ .  
Notice that you can perform  $a_i := b_{i+1}$  and  $b_i := a_{i+1}$  on the same index  $i$ .

The number of matches is 3. It can be shown that this is the maximum answer we can achieve.

In the second test case, we can do the following to achieve a maximum of 3:

- Let's choose to remove index 5. The arrays become:  $a = [2, 1, 5, 3, 4]$ ,  $b = [3, 2, 4, 5, 6]$ .
- Choose index 4, and set  $b_4 := a_5$ . The arrays become:  $a = [2, 1, 5, 3, 4]$ ,  $b = [3, 2, 4, 4, 6]$ .
- Choose index 3, and set  $a_3 := b_4$ . The arrays become:  $a = [2, 1, 4, 3, 4]$ ,  $b = [3, 2, 4, 4, 6]$ .
- Choose index 2, and set  $a_2 := b_3$ . The arrays become:  $a = [2, 4, 4, 3, 4]$ ,  $b = [3, 2, 4, 4, 6]$ .
- Choose index 1, and set  $b_1 := a_2$ . The arrays become:  $a = [2, 4, 4, 3, 4]$ ,  $b = [4, 2, 4, 4, 6]$ .
- Choose index 2, and set  $b_2 := a_3$ . The arrays become:  $a = [2, 4, 4, 3, 4]$ ,  $b = [4, 4, 4, 4, 6]$ .
- Choose index 1, and set  $a_1 := b_2$ . The arrays become:  $a = [4, 4, 4, 3, 4]$ ,  $b = [4, 4, 4, 4, 6]$ .

In the third test case, it can be shown that we can not get any matches. Therefore, the answer is 0.

DeepL 翻译

注

在第一个测试案例中，我们可以进行以下操作：

- 我们将选择不删除任何索引。
- 选择索引 3，并设置  $a_3 := b_4$ 。这样数组就变成了  $a = [1, 3, 2, 4]$ ， $b = [4, 3, 2, 2]$ 。
- 选择索引 1，并设置  $a_1 := b_2$ 。数组变为  $a = [3, 3, 2, 4]$ ， $b = [4, 3, 2, 2]$ 。
- 选择索引 1，并设置  $b_1 := a_2$ 。数组变为  $a = [3, 3, 2, 4]$ ， $b = [3, 3, 2, 2]$ 。请注意，您可以在同一个索引  $i$  中执行  $a_i := b_{i+1}$  和  $b_i := a_{i+1}$  命令。

匹配次数为 3。可以证明，这是我们能得到的最大答案。

在第二个测试案例中，我们可以通过以下方法获得最大值 3：

- 我们选择删除索引 5。这样数组就变成了  $a = [2, 1, 5, 3, 4]$ ， $b = [3, 2, 4, 5, 6]$ 。
- 选择索引 4，并设置  $b_4 := a_5$ 。数组变为  $a = [2, 1, 5, 3, 4]$ ， $b = [3, 2, 4, 4, 6]$ 。
- 选择索引 3 并设置  $a_3 := b_4$ 。数组变为  $a = [2, 1, 4, 3, 4]$ ， $b = [3, 2, 4, 4, 6]$ 。
- 选择索引 2，并设置  $a_2 := b_3$ 。数组变为  $a = [2, 4, 4, 3, 4]$ ， $b = [3, 2, 4, 4, 6]$ 。
- 选择索引 1，并设置  $b_1 := a_2$ 。数组变为  $a = [2, 4, 4, 3, 4]$ ， $b = [4, 2, 4, 4, 6]$ 。
- 选择索引 2，并设置  $b_2 := a_3$ 。数组变为  $a = [2, 4, 4, 3, 4]$ ， $b = [4, 4, 4, 4, 6]$ 。
- 选择索引 1，并设置  $a_1 := b_2$ 。数组变为  $a = [4, 4, 4, 3, 4]$ ， $b = [4, 4, 4, 4, 6]$ 。

在第三个测试案例中，我们无法得到任何匹配结果。因此，答案为 0。

GNU G++17 7.3.0

1