

「整列競争」 / 難易度 : 3

問題タイプ:コーディング問題 目標タイム:30分 アルゴリズム/ アイディア

問題文

整列競争という独自のゲームについて考えます。

整列競争に参加する1チームの人数は $2N$ 人であり、それぞれ各自 $1 \sim 2N$ の番号カードを1枚ずつバラバラに持っています。そして、各自初期位置に従って、東西に1列に並びます。

整列競争は、西から N 人全員の番号カードが $1 \sim N$ のどれか、東から N 人全員の番号カードが $N + 1 \sim 2N$ のどれかとなったとき、ゲーム終了の条件を満たし、クリアとなります。

そして、整列競争に参加するチームは以下の位置替えを何度でも行うことができます。

- 西から N 人の中で一人を選ぶ。
- 東から N 人の中で一人を選ぶ。
- 選んだ二人の位置を交換する。

整列競争が終了した時、位置替えの回数が最小のチームが勝利となります。

あなたのチームは $2N$ 人であり、東西に渡って一列に並び、西から i 番目の人は $a_i(1 \leq i \leq 2N)$ の番号カードを持っています。

あなたのチームが勝利のために最善を尽くす時、最小の位置替えの回数を出力してください。

入力される値

N
 $a_1 \ a_2 \ \cdots \ \cdots \ a_{2N-1} \ a_{2N}$

- 一行目に、チームの半分の人数を表す整数 N が与えられる。
- 二行目に、西から i 番目の人が持っている番号カードの値を表す、 $2N$ 個の整数 a_i が与えられる。

但し、 a_i は a_i を表す。

期待される出力値

あなたのチームが勝利のために最善を尽くす時の最小の位置替えの回数を一行で出力してください。

制約

- N, a_i は整数
- $N(1 \leq N \leq 10^5)$
- $a_i(1 \leq a_i \leq 2N)$

但し、番号カードは、 $1 \sim 2N$ の値のカードが一つずつ存在することが保証される。

サンプルケース1

入力値

3
3 6 5 1 2 4

期待される出力値

サンプルケース1

入力値 行数: 3

3
3 6 5 1 2 4

出力値 行数: 2

2

サンプルケース2

入力値 行数: 3

4
8 6 5 7 4 1 2 3

出力値 行数: 2

4

サンプルケース3

入力値 行数: 3

5
4 3 2 5 1 8 7 6 9 10

出力値 行数: 2

0

テストする

新しい問題の追加

2

説明

あなたのチームは3 × 2人のチームであり、西から一列に並んでいます。初期状態で一番西にいる人は番号カード3を持っており、一番東にいる人は番号カード4を持っています。

初期状態において、一番西にいる人は1 ~ 3のうち、3のカードを持っているため誰かと交代する必要がありません。一番東にいる人は4 ~ 6のうち、4のカードを持っているため誰かと交代する必要がありません。

しかし、西から2, 3, 4, 5番目の人は条件を満たさないため誰かと位置替えをする必要があります。以下のような位置替えをすると、最小回数でゲーム終了を迎えることができます。

1. (3, 6, 5, 1, 2, 4)(初期状態)

2. (3, 2, 5, 1, 6, 4)(西から2番目の人と、5番目の人が位置替え)

3. (3, 2, 1, 5, 6, 4)(西から3番目の人と、4番目の人が位置替え)

よって、位置替えの最小回数は2回となります。他にも最小回数となる位置替えの方法は存在します。

サンプルケース2

入力値

4

8 6 5 7 4 1 2 3

期待される出力値

4

説明

あなたのチームは4 × 2人のチームであり、西から一列に並んでいます。初期に一番西にいる人は番号カード8を持っており、一番東にいる人は番号カード3を持っています。

初期状態において、位置替えをする必要のない人が一人もいません。また、位置替えの最小回数は4回となります。

サンプルケース3

入力値

5

4 3 2 5 1 8 7 6 9 10

期待される出力値

0

説明

あなたのチームは5 × 2人のチームであり、西から一列に並んでいます。初期に一番西にいる人は番号カード4を持っており、一番東にいる人は番号カード10を持っています。

初期状態において、全員位置替えをする必要がありません。よって、位置替えの最小回数は0回となります。

配置変更

C++



```
1 #include<bits/stdc++.h>
2
3 using namespace std;
4
```

```
4
5  int main(){
6      cout << 0 << endl;
7      return(0);
8  }
```

採点開始

2017 444 Inc. all rights reserved