



Arranging Shoes

O Adnan é dono da maior loja de sapatos em Baku. Uma caixa que contém n pares de sapatos acabou de chegar à loja. Cada caixa consiste em dois pares de sapatos do mesmo tamanho: um para o pé esquerdo e outro para o pé direito. O Adnan colocou todos os $2n$ sapatos numa linha que consiste em $2n$ **posições** numeradas de 0 a $2n - 1$ da esquerda para a direita.

O Adnan quer reposicionar os sapatos numa **disposição válida**. Uma disposição é válida se e só se para todos os i ($0 \leq i \leq n - 1$), as condições seguintes verificam-se:

- Os sapatos nas posições $2i$ e $2i + 1$ são do mesmo tamanho.
- O sapato na posição $2i$ é um sapato para o pé esquerdo.
- O sapato na posição $2i + 1$ é um sapato para o pé direito.

Para isto, o Adnan pode fazer uma série de trocas. Em cada troca, ele escolhe dois sapatos que são **adjacentes** nesse momento e troca-os (isto é, pega neles e põe cada um na posição antiga do outro). Dois sapatos são adjacentes se as suas posições diferirem por um.

Determina o número mínimo de trocas que o Adnan precisa de fazer para obter uma disposição válida dos sapatos.

Detalhes de implementação

Deves implementar a seguinte função:

```
int64 count_swaps(int[] S)
```

- S : é uma array de $2n$ inteiros. Para cada i ($0 \leq i \leq 2n - 1$), $|S[i]|$ é um valor não nulo igual ao tamanho do sapato inicialmente colocado na posição i . Aqui, $|x|$ indica o valor absoluto de x , que é igual a x se $x > 0$ e é igual $-x$ se $x < 0$. Se $S[i] < 0$, o sapato na posição i é do pé esquerdo; caso contrário, é do pé direito.
- Esta função deve devolver o número mínimo de trocas (de sapatos adjacentes) que precisam de ser feitas de modo a obter uma disposição válida.

Exemplos

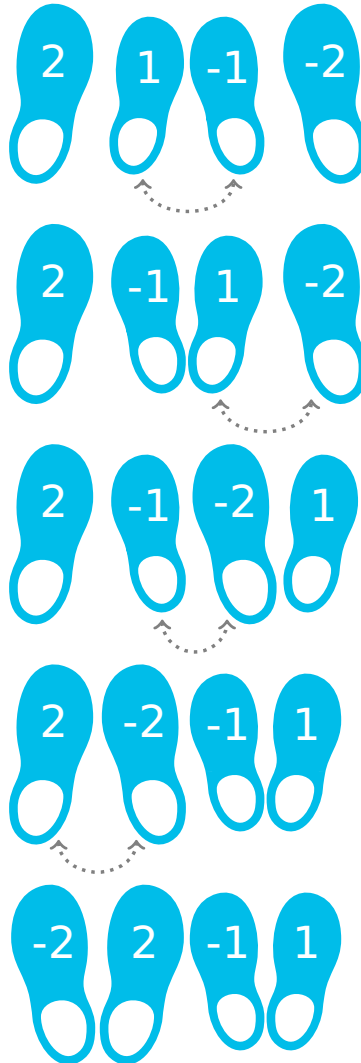
Exemplo 1

Considera a seguinte chamada:

```
count_swaps([2, 1, -1, -2])
```

O Adnan pode obter uma disposição válida fazendo 4 trocas.

Por exemplo, ele pode começar por trocar os sapatos 1 e -1 , depois 1 e -2 , depois -1 e -2 , e finalmente 2 e -2 . Ele iria então obter a seguinte disposição válida: $[-2, 2, -1, 1]$. Não é possível obter qualquer disposição válida com menos de 4 trocas. Deste modo, a função deve devolver 4.



Exemplo 2

No exemplo seguinte, todos os sapatos têm o mesmo tamanho:

```
count_swaps([-2, 2, 2, -2, -2, 2])
```

O Adnan pode trocar os sapatos nas posições 2 e 3 para obter a disposição válida $[-2, 2, -2, 2, -2, 2]$, logo a função deve devolver 1.

Restrições

- $1 \leq n \leq 100\,000$
- Para cada i ($0 \leq i \leq 2n - 1$), $1 \leq |S[i]| \leq n$.
- Uma disposição válida dos sapatos pode ser obtida fazendo uma certa sequência de trocas.

Subtarefas

1. (10 pontos) $n = 1$
2. (20 pontos) $n \leq 8$
3. (20 pontos) Todos os sapatos são do mesmo tamanho.
4. (15 pontos) Todos os sapatos nas posições $0, \dots, n - 1$ são do pé esquerdo, e todos os sapatos nas posições $n, \dots, 2n - 1$ são do pé direito. Além disso, para cada i ($0 \leq i \leq n - 1$), os sapatos nas posições i e $i + n$ são do mesmo tamanho.
5. (20 pontos) $n \leq 1000$
6. (15 pontos) Nenhuma restrição adicional.

Avaliador exemplo

O avaliador exemplo lê o input no seguinte formato:

- linha 1: n
- linha 2: $S[0] S[1] S[2] \dots S[2n - 1]$

O avaliador exemplo escreve uma única linha que contém o valor de retorno de `count_swaps`.