



Arranging Shoes

Adnan é dono da maior loja de sapatos em Baku. Uma caixa contendo n pares de sapatos acabou de chegar à loja. Cada par consiste de dois sapatos do mesmo tamanho: um esquerdo e um direito. Adnan colocou todos os $2n$ sapatos em uma linha constituída por $2n$ **posições** numeradas de 0 a $2n - 1$ da esquerda para a direita.

Adnan deseja rearrumar os sapatos em um **arranjo válido**. Um arranjo é válido se e somente se, para cada i ($0 \leq i \leq n - 1$), a seguinte condição é verdadeira:

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

Para rearrumar os sapatos Adnan pode efetuar uma série de trocas. Em cada troca, ele seleciona dois sapatos que são **adjacentes** naquele momento e troca os dois sapatos de lugar (i.e., pega os sapatos e coloca cada sapato na posição anterior do outro sapato). Dois sapatos são adjacentes se a diferença entre suas posições é igual a um.

Determine o menor número de trocas que Adnan deve realizar para obter um arranjo válido de sapatos.

Detalhes de implementação

Você deve implementar o seguinte procedimento:

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

- S : um array de $2n$ números inteiros. Para cada i ($0 \leq i \leq 2n - 1$), $|S[i]|$ é um valor diferente de zero igual ao tamanho do sapato inicialmente colocado na posição i . Aqui, $|x|$ denota o valor absoluto de x , o que é igual a x se $x > 0$ e igual a $-x$ se $x < 0$. Se $S[i] < 0$, o sapato na posição i é um sapato esquerdo; caso contrário, é um sapato direito.
- Esse procedimento deve retornar o número mínimo de trocas (de sapatos adjacentes) que devem ser realizadas para obter um arranjo válido.

Exemplos

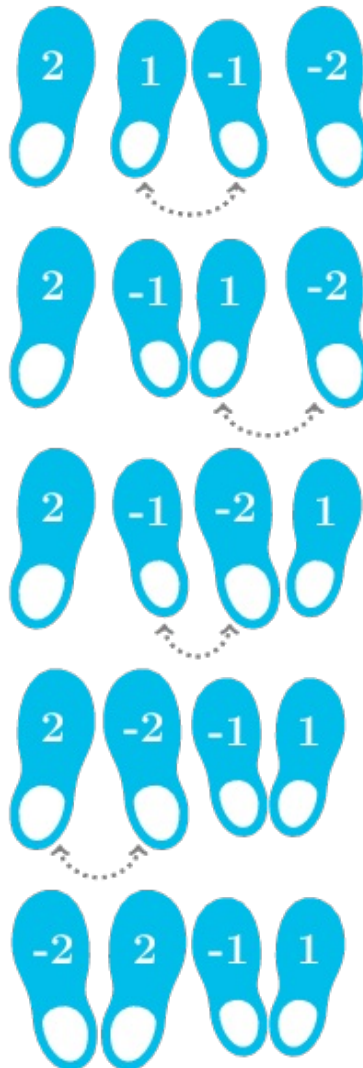
Exemplo 1

Considere a seguinte chamada:

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

Adnan pode obter um arranjo válido com 4 trocas.

Por exemplo, ele pode primeiro trocar os sapatos 1 e -1 , depois 1 e -2 , depois -1 e -2 , e finalmente 2 e -2 . Ele assim obterá o seguinte arranjo válido: $[-2, 2, -1, 1]$. Não é possível obter um arranjo válido com menos do que 4 trocas. Portanto, o procedimento deve retornar 4.



Example 2

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

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

Adnan pode trocar os sapatos nas posições 2 e 3 para obter o arranjo válido $[-2, 2, -2, 2, -2, 2]$, portanto o procedimento deve retornar 1.

Restrições

- $1 \leq n \leq 100\,000$
- Para cada i ($0 \leq i \leq 2n - 1$), $1 \leq |S[i]| \leq n$. Aqui, $|x|$ denota o valor absoluto de x .
- Um arranjo válido dos sapatos pode ser obtido realizando uma 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 sapatos esquerdos, e todos os sapatos nas posições $n, \dots, 2n - 1$ são sapatos direitos. Adicionalmente, para cada i ($0 \leq i \leq n - 1$), os sapatos nas posições i and $i + n$ são do mesmo tamanho.
5. (20 pontos) $n \leq 1000$
6. (15 pontos) Nenhuma restrição adicional.

Corretor exemplo

O corretor exemplo lê a entrada no seguinte formato:

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

O corretor exemplo escreve na saída uma única linha contendo o valor de retorno de `count_swaps`.