



# Arranging Shoes

Adnan es dueño de la tienda más grande de zapatos en Baku.

Una caja con  $n$  pares de zapatos acaba de llegar a la tienda. Cada par de de zapatos consiste en un zapato izquierdo y un zapato derecho, ambos de una misma talla. Adnan colocó todos los  $2n$  zapatos en una línea de tal forma que cada uno de los  $2n$  zapatos tiene una **posición** numerada de 0 a  $2n - 1$  de izquierda a derecha.

Adnan quiere reacomodar los zapatos en un **acomodo válido**. Un acomodo es válido si y sólo si para todo  $i$  ( $0 \leq i \leq n - 1$ ), la siguiente condición se cumple:

- Los zapatos en las posiciones  $2i$  y  $2i + 1$  son de la misma talla.
- El zapato en la posición  $2i$  es izquierdo.
- El zapato en la posición  $2i + 1$  es derecho.

Por lograr esto, Adnan puede hacer una serie de intercambios. En cada intercambio, Adnan selecciona dos zapatos que son **adyacentes** en ese momento y los intercambia (esto es, toma los dos zapatos y coloca cada uno en la posición original del otro). Dos zapatos son adyacentes si sus posiciones difieren por uno.

Determina el mínimo número de intercambios que Adnan necesita realizar para lograr un acomodo válido de zapatos.

## Detalles de implementación

Debes implementar el siguiente procedimiento:

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

- $S$ : un arreglo de  $2n$  enteros. Para cada  $i$  ( $0 \leq i \leq 2n - 1$ ),  $S[i]$  es un valor distinto de cero que describe el zapato que se encuentra inicialmente en la posición  $i$ .  $|x|$  denota el valor absoluto de  $x$ , el cual, si  $x > 0$  es  $x$  y es igual a  $-x$  si  $x < 0$ . El valor absoluto de  $S[i]$  es la talla del zapato. La talla del zapato no excede  $n$ . Si  $S[i] < 0$ , el zapato en la posición  $i$  es un zapato izquierdo; de lo contrario, es un zapato derecho.
- El procedimiento debe retornar la mínima cantidad de intercambios de zapatos adyacentes para lograr un acomodo válido.

# Ejemplos

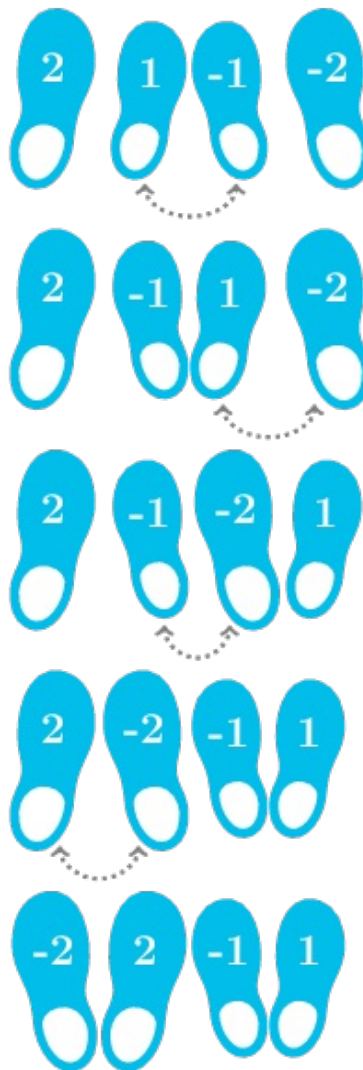
## Ejemplo 1

Considera la siguiente llamada:

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

Adnan puede lograr un acomodo válido en 4 intercambios.

Por ejemplo, él puede intercambiar los zapatos con talla 1 y  $-1$  y después los zapatos con talla 1 y  $-2$ , después los zapatos con talla  $-1$  y  $-2$  y por último los zapatos con talla 2 y  $-2$ . Él obtendrá los zapatos con el siguiente acomodo válido:  $[-2, 2, -1, 1]$ . No es posible obtener ningún acomodo válido en menos de 4 intercambios. Por lo tanto, el procedimiento debe regresar 4.



## Ejemplo 2

En el siguiente ejemplo, todos los zapatos tienen la misma talla:

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

Adnan puede intercambiar los zapatos en las posiciones 2 y 3 para obtener el acomodo válido  $[-2, 2, -2, 2, -2, 2]$ , por lo tanto el procedimiento debe regresar 1.

## Restricciones

- $1 \leq n \leq 100\,000$
- Para cada  $i$  ( $0 \leq i \leq 2n - 1$ ),  $1 \leq |S[i]| \leq n$ .
- Un acomodo válido siempre puede ser obtenido realizando alguna cantidad de intercambios adyacentes.

## Sub-tareas

1. (10 puntos)  $n = 1$
2. (20 puntos)  $n \leq 8$
3. (20 puntos) Todos los zapatos son de la misma talla.
4. (15 puntos) Todos los zapatos con posición  $0, \dots, n - 1$  son zapatos izquierdos y todos los zapatos  $n, \dots, 2n - 1$  son zapatos derechos. Además, para cada  $i$  ( $0 \leq i \leq n - 1$ ), los zapatos en las posiciones  $i$  y  $i + n$  son de la misma talla.
5. (20 puntos)  $n \leq 1000$
6. (15 puntos) Sin restricciones adicionales.

## Grader de ejemplo

El grader de ejemplo lee la entrada en el siguiente formato:

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

El grader de ejemplo imprime el valor que regresa `count_swaps`.