



# Acomodando Zapatos

Adnan es dueño de la tienda de zapatos más grande en Baku. Una caja que tiene  $n$  pares de zapatos llegó recién a la tienda. Cada par consiste de de dos zapatos del mismo tamaño: uno izquierdo y otro derecho. Adnan puso los  $2n$  zapatos en una fila que consiste de  $2n$  **posiciones** numeradas de 0 a  $2n - 1$  de izquierda a derecha.

Adnan quiere acomodar los zapatos en un **orden válido**. Un orden es válido sí y sólo sí por cada  $i$  ( $0 \leq i \leq n - 1$ ), las siguientes condiciones se cumplen:

- Los zapatos en posiciones  $2i$  y  $2i + 1$  son del mismo tamaño.
- El zapato en la posición  $2i$  es un zapato izquierdo.
- El zapato en la posición  $2i + 1$  es un zapato derecho.

Para este propósito, Adnan puede hacer los siguientes cambios. En cada cambio, él elige dos zapatos que sean **adyacentes** en ese momento y los intercambia (por ejemplo, levanta y coloca cada uno en la posición original del anterior zapato). Dos zapatos son adyacentes si sus posiciones difieren por uno.

Determine el mínimo número de cambios que Adnan necesita realizar para obtener un orden válido de los zapatos,

## Detalles de implementación

Debes implementar el siguiente procedimiento:

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

- $S$ : un arreglo de  $2n$  enteros. Por cada  $i$  ( $0 \leq i \leq 2n - 1$ ),  $|S[i]|$  es un valor distinto de cero igual al tamaño del zapato colocado inicialmente en la posición  $i$ . Aquí,  $|x|$  denota el valor absoluto de  $x$ , que es igual a  $x$  si  $x > 0$  y es igual a  $-x$  si  $x < 0$ . Si  $S[i] < 0$ , el zapato en la posición  $i$  es un zapato izquierdo; de otra manera, es un zapato derecho.
- Este procedimiento debe retornar el mínimo número de cambios (de zapatos adyacentes) que necesitan ser realizados para obtener un orden válido.

## Ejemplos

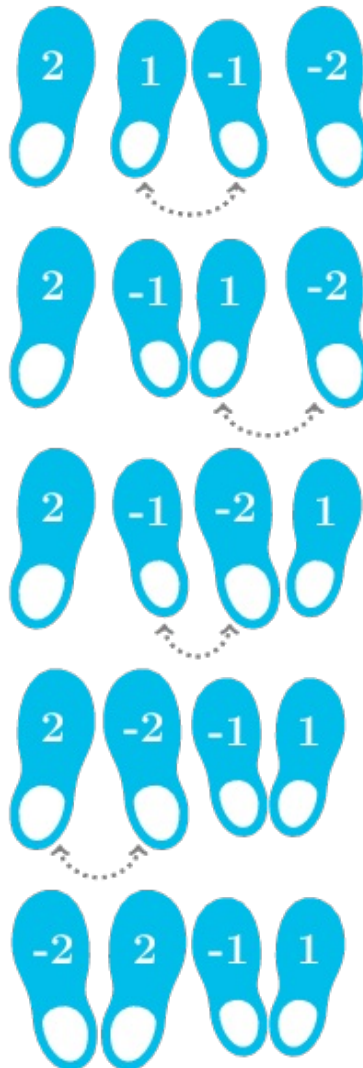
### Ejemplo 1

Considere la siguiente llamada:

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

Adnan puede obtener un orden válido en 4 cambios.

Por ejemplo, primero él puede intercambiar los zapatos 1 y  $-1$ , luego 1 y  $-2$ , luego  $-1$  y  $-2$ , y finalmente 2 y  $-2$ . Él podría luego obtener el siguiente orden válido:  $[-2, 2, -1, 1]$ . No es posible obtener ningún orden válido con menos de 4 intercambios. Por lo tanto, el procedimiento debe retornar 4.



## Ejemplo 2

En el siguiente ejemplo, todos los zapatos tienen el mismo tamaño:

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

Adnan puede intercambiar los zapatos en las posiciones 2 y 3 para obtener el orden válido  $[-2, 2, -2, 2, -2, 2]$ , entonces el procedimiento debe retornar 1.

## Restricciones

- $1 \leq n \leq 100\,000$
- Por cada  $i$  ( $0 \leq i \leq 2n - 1$ ),  $1 \leq |S[i]| \leq n$ .
- Un orden valido de los zapatos puede ser obtenido realizando algunas secuencias de cambios.

## Subtareas

1. (10 puntos)  $n = 1$
2. (20 puntos)  $n \leq 8$
3. (20 puntos) Todos los zapatos son del mismo tamaño.
4. (15 puntos) Todos los zapatos en las posiciones  $0, \dots, n - 1$  son izquierdos, y todos los zapatos en posiciones  $n, \dots, 2n - 1$  son derechos. Además, por cada  $i$  ( $0 \leq i \leq n - 1$ ), los zapatos en las posiciones  $i$  e  $i + n$  son del mismo tamaño.
5. (20 puntos)  $n \leq 1000$
6. (15 puntos) Sin restricciones adicionales.

## Grader de muestra

El grader de muestra 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 muestra devuelve una línea conteniendo el valor de retorno de `count_swaps`.