



Organizando Zapatos

Adnan posee la zapatería más grande en Bakú. Una caja que contiene n pares de zapatos acaba de llegar a la tienda. Cada par consiste de dos zapatos de la misma talla: uno izquierdo y uno derecho. Adnan ha puesto todos los $2n$ zapatos en una fila consistente de $2n$ **posiciones** numeradas de 0 hasta $2n - 1$ de la izquierda a la derecha.

Adnan quiere reorganizar los zapatos en un **arreglo válido**. Un arreglo es válido si y solamente si para cada i ($0 \leq i \leq n - 1$), se cumplen las siguientes condiciones:

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

Con este propósito, Adnan puede hacer una serie de intercambios. En cada intercambio, él selecciona dos zapatos que son **adyacentes** en ese momento y los intercambia (esto es recoge a los dos y coloca cada uno en la posición previa del otro). Dos zapatos son adyacentes si sus posiciones difieren en uno.

Determinar el número mínimo de intercambios que Adnan necesita ejecutar con el propósito de obtener un arreglo válido de zapatos.

Detalles de Implementación

Usted debe 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 a cero que describe el zapato que está inicialmente en la posición i . 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, en otro caso, es un zapato derecho.
- Este procedimiento debe devolver el número mínimo de intercambios (de zapatos adyacentes) que deben ser ejecutados en orden para obtener un arreglo válido.

Ejemplos

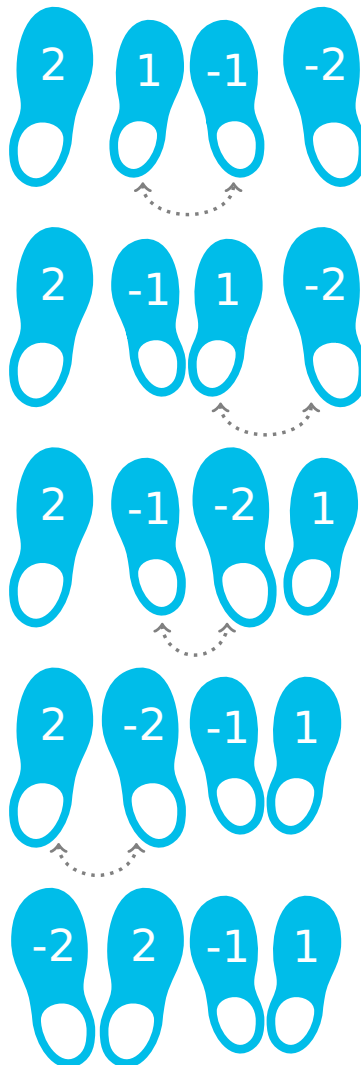
Ejemplo 1

Considere el siguiente llamado:

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

Adnan puede obtener un arreglo válido en 4 intercambios,

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

Restricciones

- $1 \leq n \leq 100\,000$
- Para cada i ($0 \leq i \leq 2n - 1$), $1 \leq |S[i]| \leq n$. Aquí, $|x|$ denota el valor absoluto de x .
- En los casos de entrada a ejecutar siempre se puede obtener un arreglo válido de zapatos ejecutando alguna sucesión de intercambios.

Subtareas

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 en las posiciones $0, \dots, n - 1$ son zapatos izquierdos, y todos los zapatos en las posiciones $n, \dots, 2n - 1$ son zapatos derechos. También, 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) No hay restricciones adicionales.

Calificador ejemplo

El calificador ejemplo lee la entrada en el formato siguiente:

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

El calificador ejemplo da como salida una sola línea conteniendo el valor entregado por `count_swaps`.