



Ordenando Zapatos

Adnan es dueño de la tienda de zapatos más grande de Baku. Una caja que contiene n pares de zapatos acaba de llegar a la tienda. Cada par consta de dos zapatos del mismo tamaño: uno izquierdo y uno derecho. Adnan ha puesto todos los $2n$ zapatos en una fila de $2n$ **posiciones** numeradas desde 0 hasta $2n - 1$, de izquierda a derecha.

Adnan quiere reordenar los zapatos en un **orden válido**. Un orden es válido si y solo 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 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 esto, Adnan puede realizar una serie de intercambios. En cada intercambio, selecciona dos zapatos que sean **adyacentes** en ese momento y los intercambia (es decir, los toma y pone cada uno en el puesto del otro). Dos zapatos son adyacentes si sus posiciones difieren en una unidad.

Tu objetivo es determinar la cantidad mínima de intercambios que Adnan necesita hacer para obtener un orden válido de los zapatos.

Detalles de implementación

Debes implementar la siguiente función:

```
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 al zapato ubicado inicialmente en la posición i . $|x|$ indica el valor absoluto de x , que es x cuando $x > 0$ y $-x$ cuando $x < 0$. Si $S[i] < 0$, el zapato en la posición i es izquierdo; de lo contrario, es un zapato derecho.
- Esta función debe retornar la cantidad mínima de intercambios (de zapatos adyacentes) que deben realizarse para obtener un orden válido.

Ejemplos

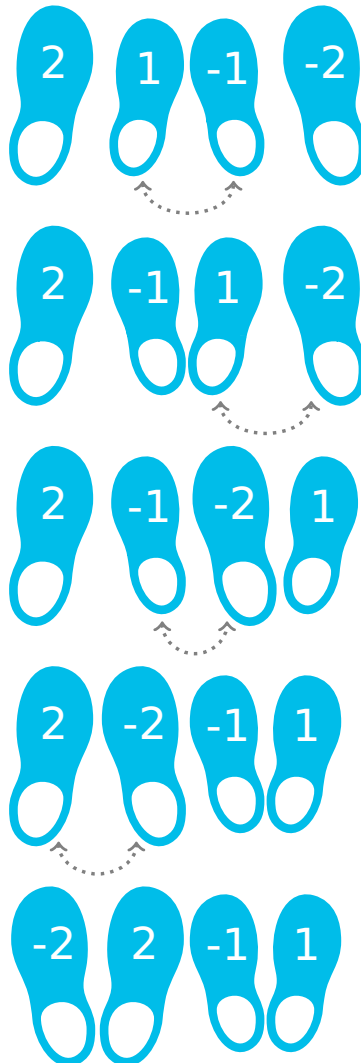
Ejemplo 1

Considera la siguiente invocación:

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

Adnan puede obtener un orden válido en 4 intercambios.

Por ejemplo, puede intercambiar los zapatos 1 y -1 , luego 1 y -2 , luego -1 y -2 , y finalmente 2 y -2 . De esta manera, obtendría el siguiente orden válido: $[-2, 2, -1, 1]$. No es posible obtener un orden válido en menos de 4 intercambios. Por lo tanto, la función 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 $[-2, 2, -2, 2, -2, 2]$, así que la función debe retornar 1.

Restricciones

- $1 \leq n \leq 100\,000$
- Para cada i ($0 \leq i \leq 2n - 1$), $1 \leq |S[i]| \leq n$.
- Siempre es posible obtener un orden válido de los zapatos realizando alguna secuencia de intercambios.

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 las posiciones $n, \dots, 2n - 1$ son derechos. Además, para 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) Ninguna restricción adicional.

Sample grader

El sample grader lee la entrada en el siguiente formato:

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

El sample grader devuelve una única línea que contiene el valor retornado por `count_swaps`.