



Disposizione di scarpe

Adnan possiede il più grande negozio di scarpe in Baku. Un pacco contenente n paia di scarpe è appena arrivato in negozio. Ogni paio di scarpe è composto da due scarpe della stessa misura: una sinistra ed una destra.

Adnan ha posizionato tutte le $2n$ scarpe in una riga consistente di $2n$ **posizioni** numerate da 0 a $2n - 1$ da sinistra a destra.

Adnan vuole sistemare le scarpe secondo una **disposizione valida**. Una disposizione si definisce valida se e solo se per ogni posizione i ($0 \leq i \leq n - 1$), le seguenti condizioni sono verificate:

- Le scarpe in posizione $2i$ e $2i + 1$ sono della stessa dimensione.
- La scarpa in posizione $2i$ è una scarpa sinistra.
- La scarpa in posizione $2i + 1$ è una scarpa destra.

Per questo compito, Adnan può effettuare una serie di scambi tra paia di scarpe in posizioni **adiacenti** (scarpe la cui posizione differisce di uno).

Determina il minor numero di scambi necessari ad Adnan al fine di ottenere una disposizione valida delle scarpe.

Dettagli di implementazione

Devi implementare la seguente funzione:

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

- S : un array di $2n$ interi. Per ogni i ($0 \leq i \leq 2n - 1$), $S[i]$ è un intero diverso da zero che descrive la scarpa inizialmente collocata in posizione i . Il valore assoluto di $S[i]$ rappresenta la dimensione della scarpa (il cui valore non supera n). Se $S[i] < 0$, la scarpa in posizione i è una scarpa sinistra; altrimenti, è una scarpa destra.
- La funzione deve restituire il minor numero di scambi (tra scarpe in posizioni adiacenti) che è necessario effettuare per ottenere una disposizione valida.

Esempi

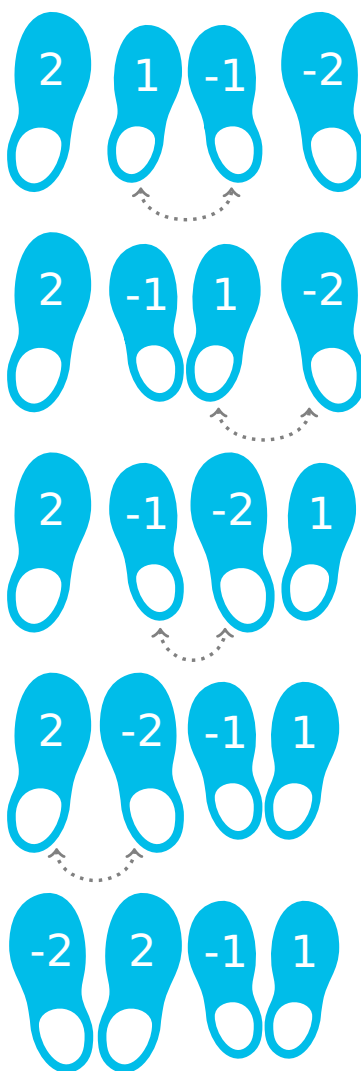
Esempio 1

Considera la seguente chiamata:

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

Adnan può ottenere una disposizione valida in 4 scambi.

Per esempio, può scambiare le scarpe 1 e -1 , poi 1 e -2 , poi -1 e -2 , infine 2 e -2 , ottenendo quindi la disposizione valida $[-2, 2, -1, 1]$. Non è possibile ottenere una disposizione valida in meno di 4 scambi, quindi la funzione deve restituire 4.



Esempio 2

Nel seguente esempio tutte le scarpe hanno la stessa dimensione:

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

Adnan può scambiare le scarpe in posizione 2 e 3 per ottenere una disposizione valida $[-2, 2, -2, 2, -2, 2]$. La funzione deve quindi restituire 1.

Assunzioni

- $1 \leq n \leq 100\,000$.
- Per ogni i ($0 \leq i \leq 2n - 1$), $1 \leq |S[i]| \leq n$, dove $|x|$ denota il *valore assoluto* di x .
- Una disposizione valida delle scarpe può sempre essere ottenuta effettuando una serie di scambi.

Subtask

1. (10 punti) $n = 1$.
2. (20 punti) $n \leq 8$.
3. (20 punti) Tutte le scarpe sono della stessa dimensione.
4. (15 punti) Tutte le scarpe in posizione $0, \dots, n - 1$ sono scarpe sinistre e tutte le scarpe in posizione $n, \dots, 2n - 1$ sono scarpe destre. Inoltre, per ogni i ($0 \leq i \leq n - 1$), le scarpe in posizione i e $i + n$ sono della stessa dimensione.
5. (20 punti) $n \leq 1000$.
6. (15 punti) Nessuna limitazione aggiuntiva.

Grader di esempio

Il grader di esempio legge l'input nel seguente formato:

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

Il grader di esempio stampa un'unica riga contenente il valore di ritorno di `count_swaps`.