



# Arranging Shoes

Adnan este proprietarul celui mai mare magazin de încălțăminte din Baku. O cutie ce conține  $n$  perechi de pantofi tocmai a sosit în magazin. Fiecare pereche este formată din doi pantofi de aceeași mărime: stânga respectiv dreapta. Adnan a pus toți cei  $2n$  pantofi într-o linie formată din  $2n$  **poziții** numerotate de la 0 la  $2n - 1$ , de la stânga la dreapta.

Adnan dorește să rearanjeze pantofii într-un **aranjament valid**. Un aranjament este valid dacă și numai dacă pentru orice  $i$  ( $0 \leq i \leq n - 1$ ), următoarele condiții sunt respectate:

- Pantofii de pe pozițiile  $2i$  și  $2i + 1$  au aceeași mărime.
- Pantoful de pe poziția  $2i$  este cel din stânga.
- Pantoful de pe poziția  $2i + 1$  este cel din dreapta.

Pentru a realiza acest lucru, Adnan poate efectua un șir de interschimbări. Într-o interschimbare, acesta selectează doi pantofi **adiacenți** în acel moment și îi interschimbă (îi ridică și pune fiecare pantof pe locul celuilalt). Doi pantofi sunt adiacenți dacă diferența absolută a pozițiilor este 1.

Determinați numărul minim de interschimbări ce Adnan trebuie să facă pentru a obține un aranjament valid de pantofi.

## Detalii de implementare

Trebuie să implementați următoarea funcție:

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

- $S$ : un vector cu  $2n$  numere întregi. Pentru fiecare  $i$  ( $0 \leq i \leq 2n - 1$ ),  $S[i]$  este o valoare nenulă ce descrie pantoful inițial aflat pe poziția  $i$ . Valoarea absolută a lui  $S[i]$  este mărimea pantofului. Mărimea pantofului nu depășește  $n$ . Dacă  $S[i] < 0$ , pantoful de pe poziția  $i$  este pantof stâng, altfel este pantof drept.
- Această funcție trebuie să returneze numărul minim de interschimbări (de pantofi adiacenți) ce trebuie efectuate pentru a obține un aranjament valid.

## Exemple

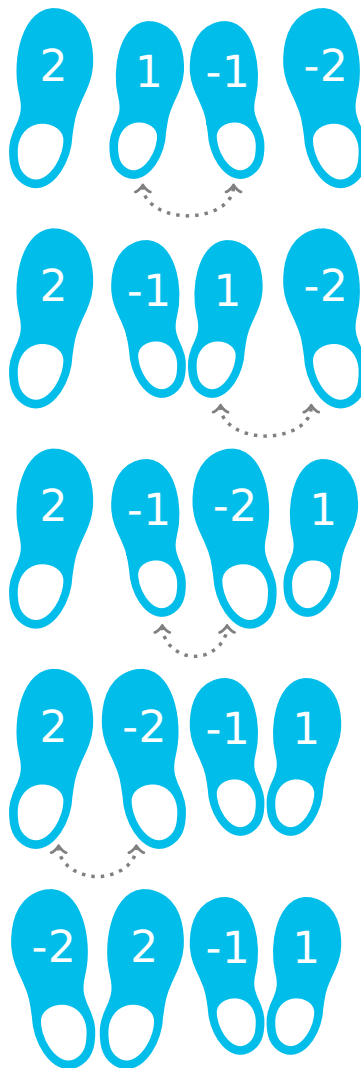
## Exemplul 1

Să considerăm următorul apel:

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

Adnan poate obține un aranjament valid în 4 interschimbări.

De exemplu, el poate face prima interschimbare între pantofii 1 și -1, apoi 1 și -2, apoi -1 și -2, și în final 2 și -2. El va obține următorul aranjament valid: [-2, 2, -1, 1]. Nu putem obține un aranjament valid în mai puțin de 4 interschimbări. Prin urmare, funcția va returna 4.



## Exemplul 2

În exemplul următor, toți pantofii au aceeași mărime:

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

Adnan poate schimba pantofii de pe pozițiile 2 și 3 pentru a obține un aranjament valid  $[-2, 2, -2, 2, -2, 2]$ , deci funcția va returna 1.

## Restricții

- $1 \leq n \leq 100\,000$
- Pentru fiecare  $i$  ( $0 \leq i \leq 2n - 1$ ),  $1 \leq |S[i]| \leq n$ .  $|x|$  reprezintă valoarea absolută a lui  $x$ .
- Se poate obține un aranjament valid al pantofilor prin efectuarea unor secvențe de interschimbări.

## Subtaskuri

1. (10 puncte)  $n = 1$
2. (20 de puncte)  $n \leq 8$
3. (20 de puncte) Toți pantofii sunt de aceeași mărime.
4. (15 puncte) Toți pantofii de pe pozițiile  $0, \dots, n - 1$  sunt de stânga, iar toți pantofii de pe pozițiile  $n, \dots, 2n - 1$  sunt de dreapta. De asemenea, pentru fiecare  $i$  ( $0 \leq i \leq n - 1$ ), pantofii de pe pozițiile  $i$  și  $i + n$  sunt de aceeași mărime.
5. (20 de puncte)  $n \leq 1000$
6. (15 puncte) Nu există alte restricții.

## Exemplu de grader

Grader-ul citește datele de intrare în formatul următor:

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

Grader-ul returnează o singură linie ce conține valoarea returnată de funcția `count_swaps`.