Réarrangement de Chaussures (Arranging Shoes)

Adnan possède le plus grand magasin de chaussures de Bakou. Une boîte contenant n paires de chaussures vient d'arriver au magasin. Chaque paire consiste en deux chaussures de la même taille : une chaussure gauche et une chaussure droite. Adnan a placé les $2n$ chaussures sur une ligne de $2n$ **positions** numérotées de 0 à $2n-1$ de gauche à droite.

Adnan souhaite réarranger les chaussures en un **arrangement valide**. Un arrangement est valide si et seulement si pour tout i ($0 \leq i \leq n-1$), les conditions suivantes sont satisfaites :

- Les chaussures aux positions $2i$ et $2i+1$ font la même taille.
- La chaussure en position $2i$ est une chaussure gauche.
- La chaussure en position $2i+1$ est une chaussure droite.

Pour ce faire, Adnan peut effectuer une séquence d'échanges. Lors de chaque échange, il choisit deux chaussures qui sont **adjacentes** à cet instant et échange leurs positions (i.e., il prend les deux chaussures et pose chaque chaussure à la position qu'occupait l'autre). Deux chaussures sont adjacentes si leur positions diffèrent de un.

Déterminez le nombre minimum d'échanges qu'Adnan doit effectuer afin d'obtenir un arrangement valide des chaussures.

Détails d'implémentation

Vous devez implémenter la fonction suivante :

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

- S: un tableau de $2n$ entiers. Pour chaque i ($0 \leq i \leq 2n-1$), $|S[i]|$ est un entier non nul égal à la taille de la chaussure placée initialement en position i. Ici, $|x|$ est la valeur absolue de x, qui vaut x si $x > 0$, et $-x$ si $x < 0$. Si $S[i] < 0$, la chaussure à la position i est une chaussure gauche ; sinon c'est une chaussure droite.
- Cette fonction doit retourner le nombre minimum d'échanges (de chaussures adjacentes) nécessaires afin d'obtenir un arrangement valide.

Exemples

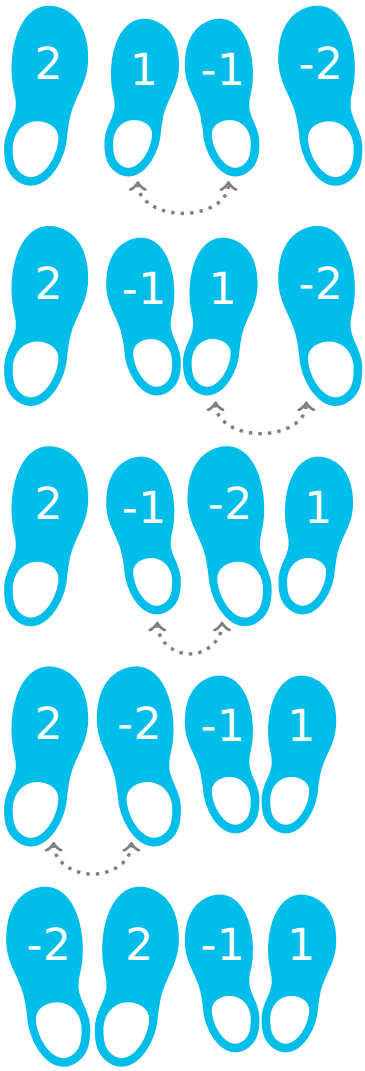
Exemple 1

On considère l'appel de fonction suivant :

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

Adnan peut obtenir un arrangement valide en effectuant 4 échanges.

Par exemple, il peut commencer par échanger les chaussures 1 et -1, puis les chaussures 1 et -2, puis -1 et -2, et enfin 2 et -2. Il obtient ainsi un arrangement valide : $[-2, 2, -1, 1]$. Il est impossible d'obtenir un arrangement valide en effectuant moins de 4 échanges. Par conséquent, la fonction doit renvoyer 4.



Example 2

Dans l'exemple suivant, toutes les chaussures ont la même taille :

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

Adnan peut échanger les chaussures en positions 2 et 3 et obtenir l'arrangement valide $[-2, 2, -2, 2, -2, 2]$, la fonction doit donc renvoyer 1.

Constraintes

- $1 \leq n \leq 100\,000$
- Pour tout i ($0 \leq i \leq 2n - 1$), $1 \leq |S[i]| \leq n$.
- Un arrangement valide des chaussures peut toujours être obtenu en effectuant une séquence d'échanges.

Sous-tâches

1. (10 points) $n = 1$
2. (20 points) $n \leq 8$
3. (20 points) Toutes les chaussures font la même taille.
4. (15 points) Les chaussures en positions $0, \ldots, n - 1$ sont des chaussures gauches, et les chaussures en positions $n, \ldots, 2n - 1$ sont des chaussures droites. De plus, pour tout i ($0 \leq i \leq n - 1$), les chaussures en positions i et $i + n$ font la même taille.
5. (20 points) $n \leq 1000$
6. (15 points) Aucune contrainte supplémentaire.

Évaluateur d'exemple (sample grader)

L'évaluateur d'exemple lit l'entrée dans le format suivant :

- ligne 1: n
- ligne 2: $S[0]$ $S[1]$ $S[2]$ $\ldots$ $S[2n - 1]$

L'évaluateur d'exemple écrit une seule ligne contenant la valeur renvoyée par la fonction `count_swaps`.