



Układanie butów

Adnan jest właścicielem największego sklepu z obuwem w Baku. Do sklepu trafiło właśnie pudełko zawierające n par butów. Każda para składa się z dwóch butów tego samego rozmiaru: lewego i prawego. Adnan ułożył wszystkie $2n$ butów w szeregu, tworząc wiersz składający się z $2n$ **pozycji** ponumerowanych od 0 do $2n - 1$ od lewej do prawej.

Adnan chce ułożyć buty we **właściwym porządku**. Porządek jest właściwy, wtedy i tylko wtedy, gdy poniższe warunki są spełnione dla każdego i ($0 \leq i \leq n - 1$):

- Buty na pozycjach $2i$ oraz $2i + 1$ mają ten sam rozmiar.
- But na pozycji $2i$ jest lewy.
- But na pozycji $2i + 1$ jest prawy.

Aby to osiągnąć, Adnan wykonuje ciąg zamian. Zamiana polega na wybraniu **sąsiednich** w danym momencie butów i zamienieniu ich miejscami (tzn. podniesieniu ich obu i położeniu każdego z nich na pozycji tego drugiego). Dwa buty są sąsiednie, jeśli ich pozycje różnią się o jeden.

Wyznacz minimalną liczbę zamian potrzebnych do ułożenia butów we właściwym porządku.

Szczegóły implementacyjne

Twoim zadaniem jest napisanie następującej funkcji:

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

- S : tablica zawierająca $2n$ liczb całkowitych. Dla każdego i ($0 \leq i \leq 2n - 1$), $S[i]$ jest niezerową wartością opisującą but znajdujący się na początku na pozycji i . Wartość bezwzględna liczby $S[i]$ to rozmiar tego buta. Jeśli $S[i] < 0$, to but na pozycji i jest lewy; jeśli $S[i] > 0$, to jest prawy.
- Funkcja ta powinna przekazać (zwrócić) minimalną liczbę zamian (sąsiednich) butów konieczną do uzyskania właściwego porządku.

Przykłady

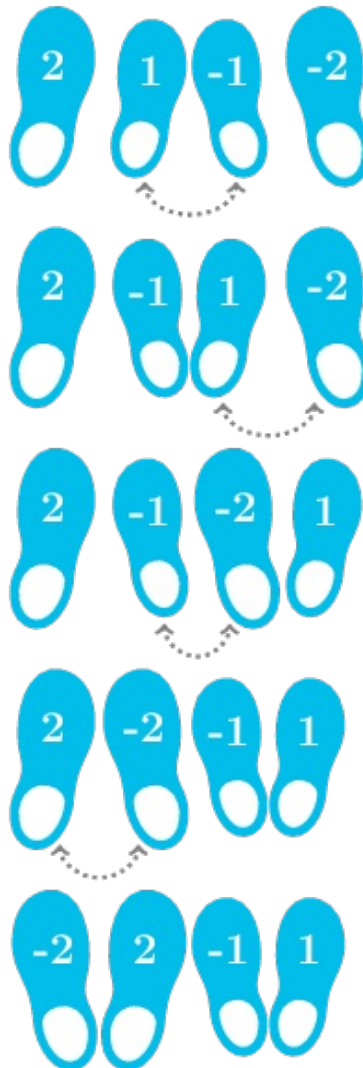
Przykład 1

Rozważ następujące wywołanie

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

Adnan może uzyskać właściwy porządek za pomocą 4 zamian.

Przykładowo może zamienić buty 1 i -1, potem 1 i -2, następnie -1 i -2, i w końcu 2 i -2. Uzyska wtedy końcowy właściwy porządek: [-2, 2, -1, 1]. Nie da się uzyskać żadnego właściwego porządku za pomocą mniej niż 4 zamian. Zatem wynikiem wywołania funkcji powinno być 4.



Przykład 2

W kolejnym przykładzie wszystkie buty są jednakowych rozmiarów:

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

Adnan może zamienić buty na pozycjach 2 i 3, aby otrzymać właściwy porządek [-2, 2, -2, 2, -2, 2], zatem wynikiem wywołania funkcji powinno być 1.

Ograniczenia

- $1 \leq n \leq 100\,000$
- Dla każdego i ($0 \leq i \leq 2n - 1$), $1 \leq |S[i]| \leq n$. Przez $|x|$ oznaczamy wartość bezwzględnej liczby x .
- Dane gwarantują, że możliwe jest uzyskanie właściwego porządku za pomocą ciągu zamian.

Podzadania

1. (10 punktów) $n = 1$
2. (20 punktów) $n \leq 8$
3. (20 punktów) Wszystkie buty są tego samego rozmiaru.
4. (15 punktów) Wszystkie buty na pozycjach $0, \dots, n - 1$ są lewe, a pozostałe na pozycjach $n, \dots, 2n - 1$ są prawe. Równocześnie, dla każdego i ($0 \leq i \leq n - 1$), buty na pozycjach i oraz $i + n$ mają ten sam rozmiar.
5. (20 punktów) $n \leq 1000$
6. (15 punktów) Bez dodatkowych ograniczeń.

Przykładowa sprawdzaczka

Przykładowa sprawdzaczka czyta dane wejściowe w następującym formacie:

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

Przykładowa sprawdzaczka wypisuje pojedynczy wiersz zawierający wynik wywołania `count_swaps`.