



Slaganje cipela

Adnan je vlasnik najveće trgovine obuće u Bakuu. Pošiljka sa n pari cipela upravo je stigla u prodavnicu. Svaki par cipela se sastoji od dvije cipele iste veličine: lijeve i desne. Adnan je sve cipele, njih $2n$, složio u red koji se sastoji od $2n$ **pozicija** numerisanih brojevima od 0 do $2n - 1$ s lijeva na desno.

Adnan želi presložiti cipele u **validan redosljed**. Redosljed je validan ako i samo ako za svaki i ($0 \leq i \leq n - 1$) vrijede sljedeći uvjeti:

- Cipele na pozicijama $2i$ i $2i + 1$ su iste veličine.
- Cipela na poziciji $2i$ je lijeva cipela.
- Cipela na poziciji $2i + 1$ je desna cipela.

U tu svrhu Adnan može izvršiti niz zamjena. U svakoj zamjeni on odabere dvije cipele koje su u tom trenutku **susjedne** i izmjenjuje njihove pozicije (tj. uzima ih i postavlja svaku na prijašnju poziciju one druge cipele). Dvije cipele su susjedne ako se njihove pozicije razlikuju za jedan.

Odredite najmanji broj zamjena koje Adnan treba obaviti kako bi dobio validan redosljed cipela.

Detalji implementacije

Trebate implementirati sljedeću proceduru:

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

- S : niz sa $2n$ cijelih brojeva. Za svaki i ($0 \leq i \leq 2n - 1$), $S[i]$ je nulta vrijednost koja opisuje cipelu koja se prvobitno nalazi na poziciji i . Apsolutna vrijednost $S[i]$ je veličina cipele. Veličina neke cipele nikada ne prelazi n . Ako je $S[i] < 0$, cipela na položaju i je lijeva cipela; inače je desna cipela.
- Ova procedura treba vratiti minimalni broj zamjena (susjednih cipela) koje je potrebno obaviti kako bi se dobio jedan validan redosljed.

Primjeri

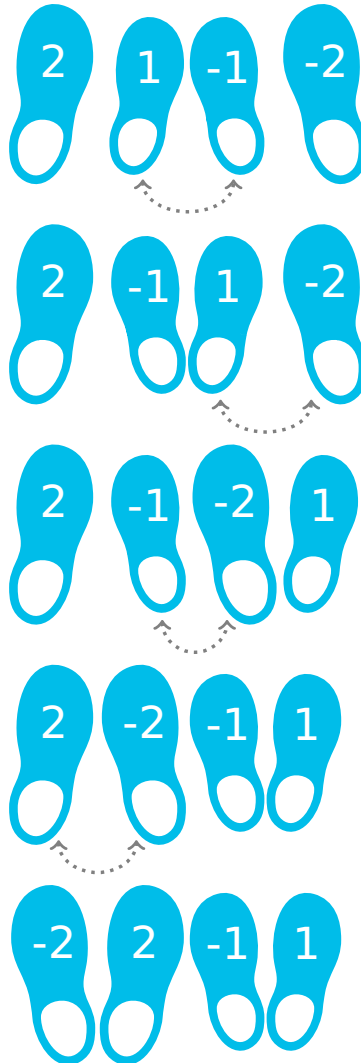
Primjer 1

Posmatrajmo sljedeći poziv procedure:

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

Adnan može dobiti validan redoslijed nakon ukupno 4 zamjene.

Na primjer, prvo može zamijeniti cipele 1 i -1 , zatim 1 i -2 , zatim -1 i -2 , te na kraju 2 i -2 . Tako bi dobio sljedeći validan redoslijed: $[-2, 2, -1, 1]$. Nije moguće dobiti neki validan redoslijed koristeći manje od 4 zamjene susjednih cipela. Prema tome procedura treba vratiti 4.



Primjer 2

U sljedećem primjeru sve su cipele iste veličine:

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

Adnan može zamijeniti cipele na pozicijama 2 i 3 kako bi dobio validan redoslijed $[-2, 2, -2, 2, -2, 2]$, tako da procedura treba vratiti broj 1.

Ograničenja

- $1 \leq n \leq 100\,000$
- Za svaki i ($0 \leq i \leq 2n - 1$), $1 \leq |S[i]| \leq n$. Ovdje $|x|$ označava apsolutnu vrijednost broja x .
- Validan redoslijed cipela se može postići obavljanjem određenog niza zamjena.

Podzadaci

1. (10 bodova) $n = 1$
2. (20 bodova) $n \leq 8$
3. (20 bodova) Sve su cipele iste veličine.
4. (15 bodova) Sve cipele na pozicijama $0, \dots, n - 1$ su lijeve cipele, a sve cipele na pozicijama $n, \dots, 2n - 1$ su desne cipele. Također, za svaki i ($0 \leq i \leq n - 1$), cipele na pozicijama i i $i + n$ su iste veličine.
5. (20 bodova) $n \leq 1000$
6. (15 bodova) Nema dodatnih ograničenja.

Grader

Grader čita ulazne podatke u sljedećem formatu:

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

Grader ispisuje na izlazu samo jednu liniju koja sadrži vraćenu vrijednost procedure `count_swaps`.