



## Ayakkabıları dizmek

Adnan, Bakü'deki en büyük ayakkabı dükkanına sahiptir. İçinde  $n$  çift ayakkabı olan bir kutu gelir dükkana. Her çift, aynı numaraya (büyüklüğe) sahip iki ayakkabıdan oluşmaktadır: bir sol bir de sağ tek. Adnan, 0'dan  $2n - 1$ 'e kadar numaralanmış,  $2n$  adet **pozisyondan** oluşan tek bir sıra üzerine bütün ayakkabıları yerleştirir.

Adnan, ayakkabıları bir **geçerli sırada** dizmek istemektedir. Bir sıra, her  $i$  ( $0 \leq i \leq n - 1$ ) için, aşağıdaki koşullar doğru ise, geçerlidir:

- $2i$  ve  $2i + 1$  pozisyonlarındaki ayakkabılar aynı büyüklüktedir.
- $2i$  pozisyonundaki ayakkabı bir sol tektir.
- $2i + 1$  pozisyonundaki ayakkabı bir sağ tektir.

Bunun için, Adnan bir miktar değiş tokuş yapabilir. Her değiş tokuşta, o anda **ardışık** olan iki tek ayakkabıyı seçer ve onların yerlerini değiştirir (yani onları alır ve her birini diğerinin bir önceki yerine koyar). Pozisyon farkları bir olan iki ayakkabı ardışıktır.

Geçerli bir sıraya sokmak için Adnan'ın yapması gereken minimum değiş tokuş sayısını bulunuz.

## Kodlama detayları

Aşağıdaki prosedürü kodlamalısınız:

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

- $S$ :  $2n$  tamsayı içeren bir dizi. Her  $i$  için ( $0 \leq i \leq 2n - 1$ ),  $|S[i]|$  sıfırdan farklı bir değerdir ve en başta pozisyon  $i$ 'ye konan ayakkabının büyüklüğünü göstermektedir. Burada  $|x|$ ,  $x$ 'in mutlak değerini göstermektedir;  $x > 0$  ise  $x$ 'e, değilse  $-x$ 'e eşittir. Ayakkabıların büyüklükleri  $n$ 'den büyük olamaz.  $S[i] < 0$  ise, pozisyon  $i$ 'deki bir sol tek,  $S[i] > 0$  ise bir sağ tektir.
- Bu prosedür, geçerli bir düzene ulaşmak için yapılması gereken minimum (ardışık ayakkabı) değiş tokuş sayısını dönmelidir.

## Örnekler

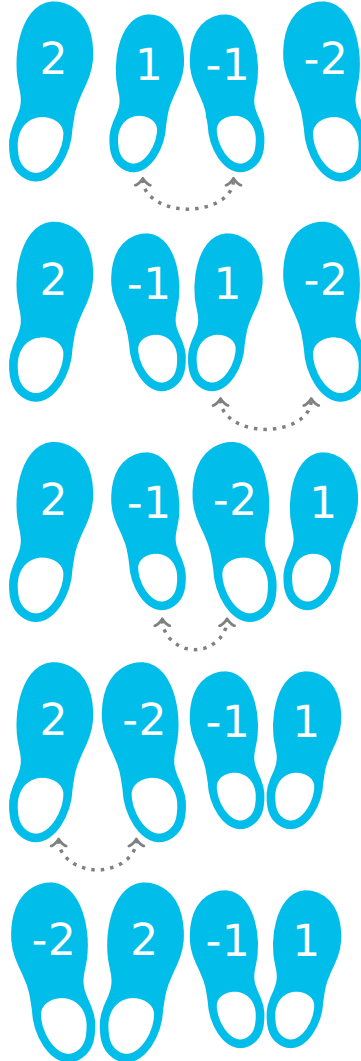
### Örnek 1

Aşağıdaki çağrı için:

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

Adnan 4 değişik tokuşta geçerli bir sıraya ulaşabilir.

Örneğin, önce 1 ve -1'i, sonra 1 ve -2'yi, sonra -1 ve -2'yi ve en sonunda 2 ve -2'yi değişik tokuş edebilir. Bunu yaparsa şu şekilde bir geçerli sıra elde edebilir: [-2, 2, -1, 1]. 4 değişik tokuştan daha az sayıda değişik tokuşla geçerli bir sıra elde etmesi mümkün değildir. Bu yüzden prosedür 4 dönmelidir.



## Örnek 2

Aşağıdaki örnekte, bütün ayakkabılar aynı numaraya (büyüklüğe) sahiptir:

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

Adnan 2 ve 3 pozisyonlarındaki ayakkabıları değişik tokuş edebilir ve [-2, 2, -2, 2, -2, 2] geçerli sırasına ulaşabilir, bu yüzden prosedür 1 dönmelidir.

## Kısıtlar

- $1 \leq n \leq 100\,000$
- Her  $i$  için ( $0 \leq i \leq 2n - 1$ ),  $1 \leq |S[i]| \leq n$ . Burada,  $|x|$ ,  $x$ 'in mutlak deęerini ifade etmektedir.
- Belli sayıda deęiş tokuş ile geęeri bir sıraya ulaşmak her zaman mümkündür.

## Altgörevler

1. (10 points)  $n = 1$
2. (20 points)  $n \leq 8$
3. (20 points) Bütün ayakkabıların numaraları (büyüklükleri) aynıdır.
4. (15 points)  $0, \dots, n - 1$  pozisyonlarındaki bütün ayakkabılar sol tek,  $n, \dots, 2n - 1$  pozisyonundakiler ise sağ tektir. Bunun yanında, her  $i$  için ( $0 \leq i \leq n - 1$ ),  $i$  ve  $i + n$  pozisyonlarındaki ayakkabıların büyüklüğü aynıdır.
5. (20 points)  $n \leq 1000$
6. (15 points) Ek kısıt bulunmamaktadır.

## Örnek puanlayıcı

Örnek puanlayıcı girdiyi aşağıdaki formatta okur:

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

Örnek puanlayıcı, `count_swaps`'in döndüğü deęeri içeren tek bir satırı çıktı olarak verir.