

Increasing Split

Zaman sınırı: 1.5 saniye

Bellek sınırı: 1024 MiB

Boris ve Ihor, N adet pozitif tam sayıdan oluşan a dizisini, yani $a_0, a_1, \dots, a_{N-1}$ dizisini bulmuşlardır ve bu diziyi kendi aralarında paylaşmak istemektedirler. Paylaşım konusunda anlaşamadıkları için, sizden arabuluculuk yapmanızı istemişlerdir.

Herhangi bir karar vermeden önce dizinin tamamı size gösterilir. Daha sonra a dizisinin elemanlarını soldan sağa doğru işlersiniz – önce a_0 , ardından a_1 ve bu şekilde a_{N-1} 'e kadar devam edersiniz. Her eleman işlendiğinde, onu Boris veya Ihor'dan tam olarak birine vermelisiniz.

Her ikisi de aldıkları elemanların, kendilerine verildikleri sırada kesin artan bir dizi oluşturmasını istemektedir. Yani, bir kişiye verdiğiniz her eleman, aynı kişiye daha önce verdiğiniz elemandan kesinlikle büyük olmalıdır. Bir kişinin aldığı ilk eleman herhangi bir değere sahip olabilir ve kişilerden birinin hiç eleman almaması da mümkündür.

Boris ve Ihor, her birinin kaç eleman alması gerektiği konusunda da anlaşmamaktadır. Bu nedenle, 0 ile N arasındaki her K tam sayısı için, elemanları Boris tam olarak K eleman alacak ve ortaya çıkan iki dizi de kesin artan olacak şekilde paylaşmanın mümkün olup olmadığını belirlemelisiniz. Her K değerini bağımsız bir soru olarak ele almalısınız: bir K değeri için çalışan bir paylaşımın başka bir K değeri için kullanılan paylaşım ile ilişkili olması gerekmez.

Örneğin, $a = [3, 1, 4, 5, 5]$ olsun ve aşağıdaki durumları ele alalım:

- $K = 3$ için: $a_0 = 3$ 'ü Boris'e, $a_1 = 1$ 'i Ihor'a, $a_2 = 4$ 'ü Boris'e, $a_3 = 5$ 'i Boris'e ve $a_4 = 5$ 'i Ihor'a verebilirsiniz. Sonuçta Boris sırasıyla 3, 4 ve 5 değerlerini, Ihor ise sırasıyla 1 ve 5 değerlerini alır. Ortaya çıkan her iki dizi de kesin artandır ve Boris 3 eleman almıştır, dolayısıyla $K = 3$ mümkündür.
- $K = 0$ için: Boris hiçbir şey almaz ve Ihor tüm elemanları alır. Ihor'un dizisi 3, 1 ile başlar ve bu kesin artan değildir, dolayısıyla $K = 0$ mümkün değildir.

Bu dizi için mümkün olan tek K değerleri 2 ve 3'tür.

Uygulama detayları

Gerçekleştirmeniz gereken fonksiyon şudur:

```
std::vector<bool> increasing_split(std::vector<int> a)
```

- a : N boyutunda bir dizi, sayı dizisi;

Bu fonksiyon tam olarak $N + 1$ boyutunda bir boolean vektör döndürmelidir. i indeksindeki eleman, elemanlar Boris tam olarak i tanesini alacak ve ortaya çıkan iki dizi de kesin artan olacak şekilde paylaşılabiliyorsa `true`, aksi takdirde `false` olmalıdır. Bu fonksiyon her test için tam olarak bir kez çağrılır.

Kısıtlar

- $2 \leq N \leq 4 \cdot 10^5$;
- tüm $0 \leq i < N$ için $1 \leq a_i \leq 10^9$.

Örnekler

Örnek değerlendirici girdisi	Örnek değerlendirici çıktısı
5 3 1 4 5 5	001100
4 1 2 3 4	11111

1. örneğin açıklaması

Yukarıdaki açıklamada kullanılanla aynı dizi olan $a = [3, 1, 4, 5, 5]$ dizisine sahibiz. Her K değeri için:

- $K = 0$: açıklamada belirtildiği gibi geçerli bir paylaşım yoktur, dolayısıyla cevap 0'dır;
- $K = 1$: Boris'in tam olarak bir eleman aldığı geçerli bir paylaşımın olmadığı benzer şekilde gösterilebilir, dolayısıyla cevap 0'dır;
- $K = 2$: $a_1 = 1$ ve $a_4 = 5$ değerlerini Boris'e, $a_0 = 3$, $a_2 = 4$ ve $a_3 = 5$ değerlerini Ihor'a verebilirsiniz. Boris sırasıyla 1 ve 5 değerlerini, Ihor ise sırasıyla 3, 4 ve 5 değerlerini alır; her iki dizi de kesin artandır, dolayısıyla cevap 1'dir;
- $K = 3$: açıklamada belirtildiği gibi geçerli bir paylaşım vardır, dolayısıyla cevap 1'dir;
- $K = 4$ ve $K = 5$: geçerli bir paylaşım yoktur, dolayısıyla cevap 0'dır.

2. örneğin açıklaması

$a = [1, 2, 3, 4]$ dizisine sahibiz ve bu dizi zaten kesin artandır. Dolayısıyla elemanların Boris ve Ihor arasında nasıl paylaştırıldığına bakılmaksızın, her ikisinin aldığı dizi de kesin artan olur. Bu nedenle 0 ile 4 arasındaki her K değeri mümkündür.

Alt görevler

Alt görev	Puan	N	Ek kısıtlar
0	0	-	Örnekler.
1	10	$N \leq 18$	-
2	5	$N \leq 4 \cdot 10^5$	tüm $0 \leq i < N - 1$ için $a_i \leq a_{i+1}$.
3	5	$N \leq 4 \cdot 10^5$	$a_0 \geq \max(a_1, a_2, \dots, a_{N-1})$.
4	16	$N \leq 4 \cdot 10^5$	a , 1 ile N arasındaki tüm tam sayıların bir permütasyonudur. $a_i < a_{i+1}$ koşulunu sağlayan her $0 \leq i < N - 1$ indeksi için, $a_0, a_1, \dots, a_i$ öneki 1 ile $i + 1$ arasındaki tüm tam sayıların bir permütasyonudur.
5	21	$N \leq 5\,000$	a , 1 ile N arasındaki tüm tam sayıların bir permütasyonudur. $\max(a_0, a_1, \dots, a_i) < a_{i+1}$ koşulunu sağlayan her $0 \leq i < N - 1$ indeksi için, $a_0, a_1, \dots, a_i$ öneki 1 ile $i + 1$ arasındaki tüm tam sayıların bir permütasyonudur.
6	17	$N \leq 4 \cdot 10^5$	a , 1 ile N arasındaki tüm tam sayıların bir permütasyonudur. $\max(a_0, a_1, \dots, a_i) < a_{i+1}$ koşulunu sağlayan her $0 \leq i < N - 1$ indeksi için, $a_0, a_1, \dots, a_i$ öneki 1 ile $i + 1$ arasındaki tüm tam sayıların bir permütasyonudur.
7	16	$N \leq 5\,000$	-
8	10	$N \leq 4 \cdot 10^5$	-

Örnek değerlendirici

Girdi formatı aşağıdaki gibidir:

- satır 1: bir tam sayı – N değeri;
- satır 2: N adet tam sayı $a_0, a_1, \dots, a_{N-1}$ – dizinin elemanları.

Döndürülen vektörün boyutu $N + 1$ değilse, örnek değerlendirici WA: Returned array does not have size N+1 sonucunu verecektir. Aksi takdirde çıktı formatı aşağıdaki gibidir:

- satır 1: uzunluğu $N + 1$ olan ikili bir dize. i indeksindeki karakter, $K = i$ mümkünse 1, aksi takdirde 0'dır.