

Növekvő felosztás

Időlimit: 1.5 másodperc

Memórialimit: 1024 MiB

Boris és Ihor találtak egy a számsorozatot $(a_0, a_1, \dots, a_{N-1})$, ami N darab pozitív egész számból áll, és ezt szeretnék egymás között felosztani. Mivel nem tudnak megegyezni a felosztásról, ezért téged kértek fel, hogy legyél a döntőbírójuk.

Mielőtt bármilyen döntést hoznál, megmutatják neked a teljes számsorozatot. Ezután balról jobbra haladva dolgozod fel az a sorozat elemeit – először az a_0 -t, majd az a_1 -et, és így tovább egészen az a_{N-1} -ig. Mindegyik elem feldolgozása után azt vagy Borisnak, vagy Ihornak kell odaadnod.

Mindketten ragaszkodnak ahhoz, hogy az általuk kapott elemek a megkapásuk sorrendjében szigorúan növekvő sorozatot alkossanak. Vagyis minden elemnek, amelyet egy személynek adsz, szigorúan nagyobb kell lennie, mint az előző elem, amelyet neki adtál. Az első elem, amelyet egy személy kap, bármilyen értékű lehet, és lehetséges, hogy egyikük egyáltalán nem kap elemet.

Boris és Ihor abban sem értenek egyet, hogy melyiküknek hány elemet kellene kapnia. Ezért minden 0 és N közötti K egész számra meg kell határozni, hogy szét lehet-e osztani az elemeket úgy, hogy Boris pontosan K elemet kapjon, és mindkét kapott sorozat szigorúan növekvő legyen. A feladatot minden egyes K értékre a többitől független feladatként kell kezelned, ami azt jelenti, hogy az $N + 1$ felosztási folyamat teljesen független egymástól.

Például legyen $a = [3, 1, 4, 5, 5]$, és tekintsük a következő eseteket:

- $K = 3$ esetén: adhatjuk $a_0 = 3$ -at Borisnak, $a_1 = 1$ -et Ihornak, $a_2 = 4$ -et Borisnak, $a_3 = 5$ -öt Borisnak és $a_4 = 5$ -öt Ihornak. Így Boris a 3, 4 és 5 számokat kapja meg ebben a sorrendben, Ihor pedig az 1 és 5 számokat kapja meg ebben a sorrendben. Mindkét kapott sorozat szigorúan növekvő, és Boris 3 elemet kapott, így $K = 3$ lehetséges.
- $K = 0$ esetén: Boris nem kap semmit, Ihor pedig mindent megkap. Így Ihor sorozata 3, 1-gyel kezdődne, ami nem szigorúan növekvő, így $K = 0$ lehetetlen.

Ennél a példánál csak $K = 2$ vagy 3 esetén lehetséges a megfelelő felosztás elkészítése.

Megvalósítás

A megvalósítandó függvény:

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

- a : N elemű számsorozat;

Ennek a függvénynek egy pontosan $N + 1$ elemű logikai értékekből álló tömböt kell visszaadnia. Az i . elemnek `true`-nak kell lennie, ha a számsorozat szétosztható úgy, hogy Boris pontosan i elemet kapjon közülük, és a két kapott számsorozat szigorúan növekvő legyen, egyébként pedig `false`-nak. Ezt a függvényt minden tesztesetnél pontosan egyszer hívják meg.

Korlátok

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

Példák

Példa bemenet	Példa kimenet
5 3 1 4 5 5	001100
4 1 2 3 4	11111

Magyarázat az 1. példához

Legyen $a = [3, 1, 4, 5, 5]$, vagyis ugyanaz a sorozat, mint a szövegben említett példában. A K egyes értékei esetén:

- $K = 0$: amint azt a feladatban is kifejtettük, nincs érvényes felosztás, így a válasz 0;
- $K = 1$: hasonlóképpen belátható, hogy nincs olyan érvényes felosztás, amelyben Boris pontosan egy elemet kapna, így a válasz 0;
- $K = 2$: Borisnak adhatjuk az $a_1 = 1$ és az $a_4 = 5$ elemeket, Ihor pedig az $a_0 = 3$, az $a_2 = 4$ és az $a_3 = 5$ elemeket kapja. Boris ebben a sorrendben kapja meg az 1-et és az 5-öt, Ihor pedig ebben a sorrendben kapja meg a 3-at, a 4-et és az 5-öt; mindkét sorozat szigorúan növekvő, így a válasz 1;
- $K = 3$: amint azt a feladatléírás is kifejti, létezik érvényes felosztás, így a válasz 1;
- $K = 4$ és $K = 5$: nem létezik érvényes felosztás, így a válasz 0.

Magyarázat a 2. példához

Adott $a = [1, 2, 3, 4]$, amely már eleve szigorúan növekvő. Ezért függetlenül attól, hogy az elemek hogyan oszlanak meg Boris és Ihor között, mindkettőjük szigorúan növekvő sorozatot kap. Így a felosztás 0-tól 4-ig minden K értékre lehetséges.

Részfeladatok

Részfeladat	Pontszám	N	További korlátok
0	0	-	A példák.
1	10	$N \leq 18$	-
2	5	$N \leq 4 \cdot 10^5$	$a_i \leq a_{i+1}$ minden $0 \leq i < N - 1$ esetén.
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 az összes 1 és N közötti egész szám permutációja. Minden $0 \leq i < N - 1$ indexre, amelyre $a_i < a_{i+1}$, az $a_0, a_1, \dots, a_i$ részsorozat az összes 1 és $i + 1$ közötti egész szám permutációja.
5	21	$N \leq 5\,000$	a az összes 1 és N közötti egész szám permutációja. Minden olyan $0 \leq i < N - 1$ indexre, amelyre $\max(a_0, a_1, \dots, a_i) < a_{i+1}$, az $a_0, a_1, \dots, a_i$ részsorozat az összes 1 és $i + 1$ közötti egész szám permutációja.
6	17	$N \leq 4 \cdot 10^5$	a az összes 1 és N közötti egész szám permutációja. Minden olyan $0 \leq i < N - 1$ indexre, amelyre $\max(a_0, a_1, \dots, a_i) < a_{i+1}$, az $a_0, a_1, \dots, a_i$ részsorozat az összes 1 és $i + 1$ közötti egész szám permutációja.
7	16	$N \leq 5\,000$	-
8	10	$N \leq 4 \cdot 10^5$	-

Mintaértékelő

A bemenet formátuma a következő:

- 1. sor: egy egész szám – az N értéke;
- 2. sor: N egész szám, $a_0, a_1, \dots, a_{N-1}$ – a sorozat elemei.

Ha a visszaadott tömb mérete nem $N + 1$, akkor a mintaértékelő a következő üzenetet adja ki: `WA: Returned array does not have size N+1`. Egyéb esetben a kimenet formátuma a következő:

- 1. sor: $N + 1$ hosszúságú bináris karakterlánc. Az i -edik indexű karaktere 1, ha $K = i$ lehetséges, egyébként 0.