

Increasing Split

Limită de timp: 1.5 secunde

Limită de memorie: 1024 MiB

Boris și Ihor au găsit o secvență a de N întregi pozitivi, $a_0, a_1, \dots, a_{N-1}$, și vor să o împartă între ei. Neputând ajunge la un acord asupra împărțirii, te-au rugat pe tine să fii arbitrul.

Înainte de a lua orice decizie, îți este prezentată întreaga secvență. Apoi procesezi elementele lui a de la stânga la dreapta – mai întâi a_0 , apoi a_1 , și așa mai departe până la a_{N-1} . Pe măsură ce fiecare element este procesat, trebuie să îl atribui exact unuia dintre Boris și Ihor.

Amândoi insistă ca elementele pe care le primesc să formeze un șir strict crescător în ordinea în care le sunt atribuite. Cu alte cuvinte, fiecare element pe care îl dai unei persoane trebuie să fie strict mai mare decât elementul precedent pe care i l-ai dat aceleiași persoane. Primul element primit de o persoană poate avea orice valoare și este posibil ca unul dintre ei să nu primească niciun element.

Boris și Ihor nu reușesc să se înțeleagă nici asupra numărului de elemente pe care ar trebui să le primească fiecare. Prin urmare, pentru fiecare număr întreg K de la 0 la N , trebuie să determini dacă este posibil să împarți elementele astfel încât Boris să primească exact K elemente, iar cele două secvențe rezultate să fie strict crescătoare. Trebuie să tratezi fiecare valoare a lui K ca pe o întrebare independentă, care are semnificația că toate $N + 1$ procese de împărțire sunt absolut independente unele de altele.

De exemplu, fie $a = [3, 1, 4, 5, 5]$ și să considerăm următoarele cazuri:

- pentru $K = 3$: îi poți da $a_0 = 3$ lui Boris, $a_1 = 1$ lui Ihor, $a_2 = 4$ lui Boris, $a_3 = 5$ lui Boris, și $a_4 = 5$ lui Ihor. La sfârșit, Boris primește 3, 4 și 5 în această ordine și Ihor primește 1 și 5, în această ordine. Ambele secvențe rezultate sunt strict crescătoare și Boris a primit 3 elemente, deci $K = 3$ este posibil.
- pentru $K = 0$: Boris nu primește nimic, iar Ihor primește totul. Secvența lui Ihor începe cu 3, 1, care nu este strict crescător deci $K = 0$ este imposibil.

Pentru acest exemplu, singurele valori valide posibile ale lui K sunt 2 și 3.

Detalii de implementare

Funcția pe care trebuie să o implementezi este:

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

- a : secvența de numere de dimensiune N ;

Această funcție trebuie să returneze un vector boolean de dimensiune exact $N + 1$. Elementul de la indexul i trebuie să fie `true` dacă elementele pot fi împărțite astfel încât Boris să primească exact i dintre ele și cele două secvențe rezultate să fie strict crescătoare, și `false` în caz contrar. Această funcție este apelată exact o dată per test.

Restricții

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

Exemple

Sample grader input	Sample grader output
5 3 1 4 5 5	001100
4 1 2 3 4	11111

Explicație pentru of exemplul 1

Avem $a = [3, 1, 4, 5, 5]$, aceeași secvență folosită în enunțul de mai sus. Pentru fiecare valoare a K :

- $K = 0$: așa cum s-a explicat în enunț, nu există o împărțire validă, deci răspunsul este 0;
- $K = 1$: se poate arăta în mod similar că nu există o împărțire validă în care Boris să primească exact un element, deci răspunsul este 0;
- $K = 2$: puteți da $a_1 = 1$ și $a_4 = 5$ lui Boris, și $a_0 = 3$, $a_2 = 4$ și $a_3 = 5$ lui Ihor. Boris primește 1 și 5 în această ordine, Ihor primește 3, 4 și 5 în această ordine; ambele secvențe sunt strict crescătoare, deci răspunsul este 1;
- $K = 3$: așa cum s-a explicat în enunț, există o împărțire validă, deci răspunsul este 1;
- $K = 4$ și $K = 5$: nu există o împărțire validă, deci răspunsul este 0.

Explicație pentru exemplul 2

Avem $a = [1, 2, 3, 4]$, care este deja strict crescătoare. Prin urmare, indiferent de modul în care elementele sunt împărțite între Boris și Ihor, fiecare dintre ei primește o secvență strict crescătoare. Deci fiecare valoare a K de la 0 la 4 este posibilă.

Subtask-uri

Subtask	Puncte	N	Restricții suplimentare
0	0	-	Exemplele.
1	10	$N \leq 18$	-
2	5	$N \leq 4 \cdot 10^5$	$a_i \leq a_{i+1}$ pentru orice $0 \leq i < N - 1$.
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 este o permutare a numerelor întregi de la 1 la N . Pentru fiecare index $0 \leq i < N - 1$ astfel încât $a_i < a_{i+1}$, prefixul $a_0, a_1, \dots, a_i$ este o permutare a tuturor numerelor întregi de la 1 la $i + 1$.
5	21	$N \leq 5\,000$	a este o permutare a numerelor întregi de la 1 la N . Pentru orice index $0 \leq i < N - 1$ astfel încât $\max(a_0, a_1, \dots, a_i) < a_{i+1}$, prefixul $a_0, a_1, \dots, a_i$ este o permutare a tuturor numerelor întregi de la 1 la $i + 1$.
6	17	$N \leq 4 \cdot 10^5$	a este o permutare a tuturor numerelor întregi de la 1 la N . Pentru orice index $0 \leq i < N - 1$ astfel încât $\max(a_0, a_1, \dots, a_i) < a_{i+1}$, prefixul $a_0, a_1, \dots, a_i$ este o permutare a tuturor numerelor întregi de la 1 la $i + 1$.
7	16	$N \leq 5\,000$	-
8	10	$N \leq 4 \cdot 10^5$	-

Sample grader

Formatul de intrare este următorul:

- linia 1: un întreg – valoarea lui N ;
- linia 2: N întregi $a_0, a_1, \dots, a_{N-1}$ – elementele secvenței.

Dacă tabloul returnat nu are dimensiunea $N + 1$, programul de testare va afișa verdictul `WA`:
Returned array does not have size N+1.

În caz contrar, formatul de ieșire este următorul:

- linia 1: un șir binar de lungime $N + 1$. Caracterul de la poziția i este 1 dacă $K = i$ este posibil, și 0 în caz contrar.