

Aufsteigende Aufteilung

Zeit: 1.5 Sekunden

Speicherplatz: 1024 MiB

Boris und Ihor haben eine Folge a von N positiven Ganzzahlen $a_0, a_1, \dots, a_{N-1}$ gefunden und wollen diese vollständig zwischen sich aufteilen. Auf die genaue Aufteilung können sie sich allerdings nicht einigen und haben dich deswegen als Schiedsrichter hinzugezogen.

Bevor du irgendwelche Entscheidungen treffen musst, zeigen sie dir die gesamte Zahlenfolge. Im Anschluss sollst du die Zahlen in der Folge a von links nach rechts durchgehen – also zuerst a_0 , dann a_1 und so weiter bis a_{N-1} . Immer wenn du bei einem Element angelangt bist, musst du es entweder Boris oder Ihor geben.

Beide bestehen darauf, dass sie ihre Zahlen in strikt aufsteigender Reihenfolge erhalten. Das bedeutet: Jede Zahl, die du einer Person übergibst, muss strikt größer sein als die Zahl, die du dieser Person davor gegeben hast. Die erste Zahl, die eine Person erhält, kann einen beliebigen Wert haben. Es ist auch möglich, dass einer keine einzige Zahl erhält.

Boris und Ihor können sich auch nicht darauf einigen, wie viele Zahlen jeder von ihnen erhalten sollte. Deswegen sollst du für jede Ganzzahl K zwischen 0 und N entscheiden, ob es möglich ist, die Zahlen so aufzuteilen, dass Boris genau K davon erhält und die beiden entstehenden Folgen strikt aufsteigend sind. Du solltest jeden Wert von K als eigenständige Frage betrachten. Das bedeutet, dass alle $N + 1$ Aufteilungen völlig unabhängig voneinander sind.

Für die Zahlenfolge $a = [3, 1, 4, 5, 5]$ betrachten wir als Beispiel die folgenden Fälle:

- für $K = 3$: Du kannst $a_0 = 3$ an Boris geben, $a_1 = 1$ an Ihor, $a_2 = 4$ an Boris, $a_3 = 5$ an Boris und $a_4 = 5$ an Ihor. Am Ende erhält Boris die Zahlen 3, 4 und 5 in dieser Reihenfolge und Ihor die Zahlen 1 und 5 in dieser Reihenfolge. Beide entstehenden Folgen sind strikt aufsteigend und Boris hat genau 3 Zahlen erhalten. Also ist $K = 3$ möglich.
- für $K = 0$: Boris dürfte keine einzige Zahl erhalten und Ihor alle. Ihors Folge müsste mit 3, 1 beginnen. Das ist nicht strikt aufsteigend, weswegen $K = 0$ nicht möglich ist.

In diesem Beispiel sind 2 und 3 die einzigen möglichen Werte für K .

Implementierungsdetails

Du musst die folgende Funktion implementieren:

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

- a : die Zahlenfolge mit Länge N ;

Diese Funktion muss ein boolesches Array mit einer Größe von genau $N + 1$ zurückliefern. Das Element mit Index i muss `true` sein, wenn die Zahlenfolge so aufgeteilt werden kann, dass Boris genau i dieser Zahlen enthält und die beiden entstehenden Folgen strikt aufsteigend sind. Andernfalls muss das Element den Wert `false` haben. Diese Funktion wird für jeden Test genau einmal aufgerufen.

Beschränkungen

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

Beispiele

Eingabe Sample Grader	Ausgabe Sample Grader
5 3 1 4 5 5	001100
4 1 2 3 4	11111

Erklärung von Beispiel 1

Wir haben $a = [3, 1, 4, 5, 5]$, also die gleiche Zahlenfolge wie im Text oben. Wir gehen alle Werte für K durch:

- $K = 0$: Laut der Erklärung im Text existiert keine gültige Aufteilung, also ist die Antwort 0;
- $K = 1$: Auf ähnliche Weise kann gezeigt werden, dass keine gültige Aufteilung existiert, in der Boris genau ein Element erhält. Die Antwort ist also wieder 0;
- $K = 2$: Du kannst $a_1 = 1$ und $a_4 = 5$ an Boris geben sowie $a_0 = 3$, $a_2 = 4$ und $a_3 = 5$ an Ihor. Boris erhält 1 und 5 in dieser Reihenfolge, Ihor erhält in dieser Reihenfolge 3, 4 und 5. Beide Folgen sind strikt aufsteigend. Die Antwort ist also 1;
- $K = 3$: Wie oben erklärt, existiert eine gültige Aufteilung. Die Antwort ist daher 1;
- $K = 4$ und $K = 5$: Es existiert keine gültige Aufteilung, also ist die Antwort 0.

Erklärung von Beispiel 2

Wir haben $a = [1, 2, 3, 4]$, was schon eine strikt aufsteigende Zahlenfolge ist. Deswegen erhalten Boris und Ihor beide eine strikt aufsteigende Zahlenfolge, unabhängig von der Aufteilung der Zahlen. Daher kann K jeden Wert zwischen 0 und 4 annehmen.

Teilaufgaben

Teilaufgabe	Punkte	N	Zusätzliche Beschränkungen
0	0	-	Die Beispieleingaben.
1	10	$N \leq 18$	-
2	5	$N \leq 4 \cdot 10^5$	$a_i \leq a_{i+1}$ für alle $0 \leq i < N - 1$.
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 ist eine Permutation aller Ganzzahlen von 1 bis N . Für jeden Index $0 \leq i < N - 1$ mit $a_i < a_{i+1}$ ist der Präfix $a_0, a_1, \dots, a_i$ eine Permutation aller Ganzzahlen von 1 bis $i + 1$.
5	21	$N \leq 5\,000$	a ist eine Permutation aller Ganzzahlen von 1 bis N . Für jeden Index $0 \leq i < N - 1$ mit $\max(a_0, a_1, \dots, a_i) < a_{i+1}$ ist der Präfix $a_0, a_1, \dots, a_i$ eine Permutation aller Ganzzahlen von 1 bis $i + 1$.
6	17	$N \leq 4 \cdot 10^5$	a ist eine Permutation aller Ganzzahlen von 1 bis N . Für jeden Index $0 \leq i < N - 1$ mit $\max(a_0, a_1, \dots, a_i) < a_{i+1}$ ist der Präfix $a_0, a_1, \dots, a_i$ eine Permutation aller Ganzzahlen von 1 bis $i + 1$.
7	16	$N \leq 5\,000$	-
8	10	$N \leq 4 \cdot 10^5$	-

Sample grader

Das Eingabeformat ist das folgende:

- Zeile 1: Eine Ganzzahl – der Wert von N ;
- Zeile 2: N Ganzzahlen $a_0, a_1, \dots, a_{N-1}$ – die Elemente der Zahlenfolge.

Wenn das zurückgegebene Array nicht Größe $N + 1$ hat, gibt der Sample grader WA: `Returned array does not have size N+1` zurück. Andernfalls ist das Ausgabeformat das folgende:

- Zeile 1: Ein Binärstring mit Länge $N + 1$. Das i -te Zeichen in diesem String ist 1, wenn $K = i$ möglich ist, und 0 sonst.