

Rosnący podział

Limit czasu: 1,5 sekundy

Limit pamięci: 1024 MiB

Błażej i Filip znaleźli ciąg a składający się z N dodatnich liczb całkowitych, $a_0, a_1, \dots, a_{N-1}$ i chcą go podzielić między siebie. Ponieważ nie potrafią dojść do porozumienia co do podziału, poprosili Cię, abyś został ich arbitrem.

Przed podjęciem jakichkolwiek decyzji poznałeś cały ciąg. Następnie przetwarzasz elementy ciągu a po kolei – najpierw a_0 , potem a_1 , i tak dalej, aż do a_{N-1} . Podczas przetwarzania każdego elementu musisz przekazać go dokładnie jednemu z nich: Błażewi albo Filipowi.

Obaj nalegają, aby otrzymywane przez nich elementy tworzyły, w kolejności ich otrzymywania, ciąg ściśle rosnący. Oznacza to, że każdy element, który dajesz danej osobie, musi być ściśle większy niż poprzedni element, który jej dałeś. Pierwszy element otrzymany przez daną osobę może mieć dowolną wartość. Możliwe jest również, że ktoś z nich nie otrzyma żadnego elementu.

Błażej i Filip nie mogą się także zgodzić co do tego, ile elementów powinien otrzymać każdy z nich. Dlatego dla każdej liczby całkowitej K od 0 do N powinienes określić, czy możliwe jest podzielenie elementów w taki sposób, aby Błażej otrzymał dokładnie K elementów, a oba powstałe ciągi były ściśle rosnące. Powinienes potraktować każdą wartość K jako niezależne pytanie. Oznacza to, że każdy z $N + 1$ procesów podziału ciągu jest niezależny od siebie.

Na przykład niech $a = [3, 1, 4, 5, 5]$. Rozważmy następujące przypadki:

- dla $K = 3$: możesz dać $a_0 = 3$ Błażewi, $a_1 = 1$ Filipowi, $a_2 = 4$ Błażewi, $a_3 = 5$ Błażewi i $a_4 = 5$ Filipowi. W rezultacie Błażej otrzymuje 3, 4 i 5 w tej kolejności, a Filip otrzymuje 1 i 5 w tej kolejności. Oba powstałe ciągi są ściśle rosnące, a Błażej otrzymał 3 elementy, więc $K = 3$ jest możliwe.
- dla $K = 0$: Błażej nie otrzymuje niczego, a Filip otrzymuje wszystko. Ciąg Filipa zaczyna się od 3, 1, co nie jest ciągiem ściśle rosnącym, więc $K = 0$ jest niemożliwe.

Dla tego ciągu jedynymi K spełniającymi warunki są 2 i 3.

Szczegóły implementacji

Zaimplementuj następującą funkcję:

```
std::vector increasing_split(std::vector a)
```

gdzie:

- a : ciąg liczb o rozmiarze N ;

Ta funkcja musi zwrócić tablicę wartości logicznych o rozmiarze dokładnie $N + 1$. Element pod indeksem i powinien mieć wartość `true`, jeśli elementy można podzielić tak, by Błażej otrzymał dokładnie i z nich, a oba powstałe ciągi były ściśle rosnące, oraz `false` w przeciwnym przypadku. Ta funkcja jest wywoływana dokładnie raz dla każdego testu.

Ograniczenia

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

Przykłady

Wejście przykładowego programu oceniającego	Wyjście przykładowego programu oceniającego
5 3 1 4 5 5	001100
4 1 2 3 4	11111

Wyjaśnienie pierwszego przykładu

Mamy $a = [3, 1, 4, 5, 5]$, czyli ten sam ciąg co w treści zadania. Dla każdej wartości K :

- $K = 0$: jak wyjaśniono w treści zadania, prawidłowy podział nie istnieje, więc odpowiedź to 0;
- $K = 1$: można pokazać, że nie istnieje prawidłowy podział, w którym Błażej otrzymuje dokładnie jeden element, więc odpowiedź to 0;
- $K = 2$: możesz dać $a_1 = 1$ i $a_4 = 5$ Błażejowi oraz $a_0 = 3$, $a_2 = 4$ i $a_3 = 5$ Filipowi. Błażej otrzymuje 1 i 5 w tej kolejności, Filip otrzymuje 3, 4 i 5 w tej kolejności; oba ciągi są ściśle rosnące, więc odpowiedź to 1;
- $K = 3$: jak wyjaśniono w treści zadania, prawidłowy podział istnieje, więc odpowiedź to 1;
- $K = 4$ i $K = 5$: prawidłowy podział nie istnieje, więc odpowiedź to 0.

Wyjaśnienie drugiego przykładu

Mamy $a = [1, 2, 3, 4]$, który jest już ściśle rosnący. Dlatego, niezależnie od tego, jak elementy zostaną podzielone między Błażeja i Filipa, każdy z nich otrzyma ściśle rosnący ciąg. Stąd każda

wartość K od 0 do 4 jest możliwa.

Podzadania

Podzadanie	Punkty	N	Dodatkowe ograniczenia
0	0	-	Przykłady z zadania.
1	10	$N \leq 18$	-
2	5	$N \leq 4 \cdot 10^5$	$a_i \leq a_{i+1}$ dla każdego $0 \leq i < N - 1$.
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 jest permutacją liczb całkowitych od 1 do N . Dla każdego indeksu $0 \leq i < N - 1$ takiego, że $a_i < a_{i+1}$, prefiks $a_0, a_1, \dots, a_i$ jest permutacją liczb całkowitych od 1 do $i + 1$.
5	21	$N \leq 5\,000$	a jest permutacją liczb całkowitych od 1 do N . Dla każdego indeksu $0 \leq i < N - 1$ takiego, że $\max(a_0, a_1, \dots, a_i) < a_{i+1}$, prefiks $a_0, a_1, \dots, a_i$ jest permutacją liczb całkowitych od 1 do $i + 1$.
6	17	$N \leq 4 \cdot 10^5$	a jest permutacją liczb całkowitych od 1 do N . Dla każdego indeksu $0 \leq i < N - 1$ takiego, że $\max(a_0, a_1, \dots, a_i) < a_{i+1}$, prefiks $a_0, a_1, \dots, a_i$ jest permutacją liczb całkowitych od 1 do $i + 1$.
7	16	$N \leq 5\,000$	-
8	10	$N \leq 4 \cdot 10^5$	-

Przykładowy program oceniający

Format wejścia jest następujący:

- wiersz 1: jedna liczba całkowita – wartość N ;
- wiersz 2: N liczb całkowitych $a_0, a_1, \dots, a_{N-1}$ – elementy ciągu.

Jeśli zwrócona tablica nie ma rozmiaru $N + 1$, przykładowy program oceniający wypisze werdykt WA: `Returned array does not have size N+1`. W przeciwnym razie format wyjścia jest następujący:

- wiersz 1: ciąg binarny o długości $N + 1$. Znak pod indeksem i to 1, jeśli $K = i$ spełnia warunki zadania, oraz 0 w przeciwnym razie.