

Partage Croissant

Limite de temps : 1.5 secondes

Limite de mémoire : 1024 Mio

Boris et Ihor ont trouvé une suite a de N entiers strictement positifs, $a_0, a_1, \dots, a_{N-1}$, et veulent la partager entre eux. Incapables de se mettre d'accord, ils vous ont demandé d'arbitrer.

Avant de prendre toute décision, on vous montre la suite entière. Vous traitez ensuite les éléments de a de gauche à droite – d'abord a_0 , puis a_1 , et ainsi de suite jusqu'à a_{N-1} . Au moment de traiter chaque élément, vous devez le donner à exactement une des deux personnes : Boris ou Ihor.

Les deux insistent pour que les éléments qu'ils reçoivent forment une suite strictement croissante dans l'ordre de réception. C'est-à-dire que chaque élément que vous donnez à une personne doit être strictement plus grand que l'élément précédent que vous lui avez donné. Le premier élément qu'une personne reçoit peut avoir n'importe quelle valeur, et il est possible que l'une des deux personnes ne reçoive aucun élément.

Boris et Ihor ne peuvent pas non plus s'accorder sur le nombre d'éléments que chacun d'eux devrait recevoir. Par conséquent, pour chaque entier K de 0 à N , vous devez déterminer s'il est possible de partager les éléments de sorte que Boris reçoive exactement K éléments et que les deux suites résultantes soient strictement croissantes. Vous devez considérer chaque valeur de K comme une question indépendante, ce qui signifie que tous les $N + 1$ processus de partage sont entièrement indépendants les uns des autres.

Par exemple, soit $a = [3, 1, 4, 5, 5]$ et considérez les cas suivants :

- pour $K = 3$: vous pouvez donner $a_0 = 3$ à Boris, $a_1 = 1$ à Ihor, $a_2 = 4$ à Boris, $a_3 = 5$ à Boris, et $a_4 = 5$ à Ihor. Au final, Boris reçoit 3, 4 et 5 dans cet ordre, et Ihor reçoit 1 et 5, dans cet ordre. Les deux suites résultantes sont strictement croissantes et Boris a reçu 3 éléments, donc $K = 3$ est possible.
- pour $K = 0$: Boris ne reçoit rien et Ihor reçoit tout. La suite d'Ihor commence par 3, 1, ce qui n'est pas strictement croissant, donc $K = 0$ est impossible.

Pour cet exemple, les seules valeurs possibles de K sont 2 et 3.

Détails d'implémentation

La fonction que vous devez implémenter est :

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

- a : la séquence de nombres de taille N ;

Cette fonction doit renvoyer un tableau booléen de taille exactement $N + 1$. L'élément à l'index i doit être `true` si les éléments peuvent être divisés de sorte que Boris reçoive exactement i d'entre eux et que les deux séquences résultantes soient strictement croissantes, et `false` sinon. Cette fonction est appelée exactement une fois par test.

Contraintes

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

Exemples

Entrée de l'évaluateur d'exemple	Sortie de l'évaluateur d'exemple
5 3 1 4 5 5	001100
4 1 2 3 4	11111

Explication de l'exemple 1

Nous avons $a = [3, 1, 4, 5, 5]$, la même séquence utilisée dans l'énoncé ci-dessus. Pour chaque K :

- $K = 0$: comme expliqué ci-dessus, aucun partage valide n'existe, donc la réponse est 0 ;
- $K = 1$: on peut montrer de la même façon qu'aucun partage valide n'existe dans lequel Boris reçoit exactement un élément, donc la réponse est 0 ;
- $K = 2$: vous pouvez donner $a_1 = 1$ et $a_4 = 5$ à Boris, et $a_0 = 3$, $a_2 = 4$ et $a_3 = 5$ à Ihor. Boris reçoit 1 et 5 dans cet ordre, Ihor reçoit 3, 4 et 5 dans cet ordre ; les deux séquences sont strictement croissantes, donc la réponse est 1 ;
- $K = 3$: comme expliqué ci-dessus, un partage valide existe, donc la réponse est 1 ;
- $K = 4$ et $K = 5$: aucun partage valide n'existe, donc la réponse est 0.

Explication de l'exemple 2

Nous avons $a = [1, 2, 3, 4]$, qui est déjà strictement croissant. Par conséquent, peu importe comment les éléments sont divisés entre Boris et Ihor, chacun d'eux reçoit une séquence strictement croissante. Donc chaque valeur de K de 0 à 4 est possible.

Sous-tâches

Sous-tâche	Points	N	Contraintes supplémentaires
0	0	-	Exemples du sujet.
1	10	$N \leq 18$	-
2	5	$N \leq 4 \cdot 10^5$	$a_i \leq a_{i+1}$ pour chaque $0 \leq i < N - 1$.
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 est une permutation de tous les entiers de 1 à N . Pour chaque indice $0 \leq i < N - 1$ tel que $a_i < a_{i+1}$, le préfixe $a_0, a_1, \dots, a_i$ est une permutation de tous les entiers de 1 à $i + 1$.
5	21	$N \leq 5\,000$	a est une permutation de tous les entiers de 1 à N . Pour chaque indice $0 \leq i < N - 1$ tel que $\max(a_0, a_1, \dots, a_i) < a_{i+1}$, le préfixe $a_0, a_1, \dots, a_i$ est une permutation de tous les entiers de 1 à $i + 1$.
6	17	$N \leq 4 \cdot 10^5$	a est une permutation de tous les entiers de 1 à N . Pour chaque indice $0 \leq i < N - 1$ tel que $\max(a_0, a_1, \dots, a_i) < a_{i+1}$, le préfixe $a_0, a_1, \dots, a_i$ est une permutation de tous les entiers de 1 à $i + 1$.
7	16	$N \leq 5\,000$	-
8	10	$N \leq 4 \cdot 10^5$	-

Évaluateur d'exemple

Le format d'entrée est le suivant :

- ligne 1 : un entier – la valeur de N ;
- ligne 2 : N entiers $a_0, a_1, \dots, a_{N-1}$ – les éléments de la séquence.

Si le tableau renvoyé n'est pas de taille $N + 1$, l'évaluateur d'exemple affichera le verdict `WA` :

`Returned array does not have size N+1`. Sinon, le format de sortie est le suivant :

- ligne 1 : une chaîne binaire de longueur $N + 1$. Son caractère à l'index i est 1 si $K = i$ est possible, et 0 sinon.