Задача 3. Бягство

⌛0.6 sec. 💾 256 MB

Относително отвращаващите задачи за контролите за разширения младши отбор, които Боби написа, му донесоха много неприятности. Сашка реши да направи план за неговото преследване и залавяне. За да избяга последиците, Боби заживял в гора, съдържаща N дървета, наредени в права линия. Дърветата са номерирани с числата от 0 до $N-1$ спрямо позицията им, като височината на i-тото дърво отляво-надясно е равно на H_i метра. Височините на дърветата са **различни** числа от 1 до N. Иначе казано, височините $H_0, H_1, ..., H_{N-1}$ образуват пермутация на $\{1, 2, 3, ..., N\}$.

За да живее успешно в гората, Боби доби орангутанови умения. Като се е качил на дърво номер x, той може да скочи на всяко дърво номер y, за което:

- $y < x, H_y > H_x$ и за $\forall i, y < i < x, H_i < H_x$.
- $y > x, H_y > H_x$ и за $\forall i, x < i < y, H_i < H_x$.

Сашка знае навиците на Боби, заради което тя трябва да планира неговия улов. Тя иска да разгледа Q сценария, в които Боби започва серията му скоци от дърво измежду номера A и B включително и завършва в дърво с номер измежду C и D включително ($A \leq B < C \leq D$). Тя се чуди коя е най-кратката такава последователност. Напишете програма `jumps`, отговаряща на тези въпроси.

Детайли по имплементацията

Задачата е дадена в интерактивен вариант, в който Вие трябва да имплементирате две функции, които да изпълнят дадената задача.

Функцията `void init(int N, std::vector <int> H)`, която трябва да напишете, ще бъде извикана само веднъж от програмата на журито и като аргумент ще получи цялото число N, както и векторът H, където $H[i]$ ще е височината на i-тото дърво.

Функцията `int minimum_jumps(int A, int B, int C, int D)`, която трябва да напишете, ще бъде извикана Q на брой пъти от програмата на журито и като аргументи ще получи четири числа, съответно A, B, C и D за текущия въпрос. Тя трябва да върне минималния брой скоци за преминаване от някое дърво измежду номера от A до B до дърво с номера C до D. Ако няма начин Боби да стигне от дърво с номер между A и B до дърво с номер измежду C и D, функцията трябва да върне -1. Тя ще бъде извиквана след извикването на `init`.

Ограничения

- $2 \leq N \leq 200\,000$
- $1 \leq Q \leq 100\,000$
- $1 \leq H_i \leq N (0 \leq i \leq N-1)$
- $H_i \neq H_j (0 \leq i < j \leq N-1)$
- $0 \leq A \leq B < C \leq D \leq N-1$

Подзадачи

Подзадача	**Точки**	N	Q	**Други ограничения**
1	0	–	–	Примерите
2	4	$\leq 200\,000$	$\leq 200\,000$	$H_i = i + 1$
3	8	≤ 200	≤ 200	–
4	13	$\leq 2\,000$	$\leq 2\,000$	–
5	12	$\leq 200\,000$	≤ 5	–
6	23	$\leq 200\,000$	$\leq 200\,000$	$A = B, C = D$
7	21	$\leq 200\,000$	$\leq 200\,000$	$C = D$
8	19	$\leq 200\,000$	$\leq 200\,000$	–

Точките за дадена подзадача се получават само ако се преминат успешно всички тестове, предвидени за нея.

Локално тестване

Предоставен Ви е файлът `Lgrader.cpp`, който може да компилирате заедно с вашата програма, за да я тествате. При стартиране програмата ще чете от стандартния вход:

- На първия ред: N и Q
- На втория ред: $H_0, H_1, ..., H_{N-1}$.
- На останалите Q реда: A, B, C, D.

Примерен вход и изход на Lgrader

Вход	**Изход**	**Обяснение на примера**

`7 3`
`3 2 1 6 4 5 7`
`4 4 6 6`
`1 3 5 6`
`0 1 2 2`
	`2`
	`1`
	`-1`
		`Скоците са от дървета с индекси:`
		`(1)` $4 \to 3 \to 6$
		`(2)` $3 \to 6$
		`(3)` $-$