



## Problema Weirdtree

Header C++      `weirdtree.h`

Azusa, vrăjitoarea zonelor înalte, a descoperit o grădină plină de copaci ciudați! Astfel, împreună cu prietena ei, Laika, s-a decis să petreacă niște timp acolo, având grijă de grădină.

Grădina poate fi văzută ca o secvență cu  $N$  copaci, în care copacii sunt numerotați de la 1 la  $N$ . Fiecare copac are o anumită înălțime număr natural. Astfel Azusa își va petrece timpul în concordanță cu un orar ce conține  $Q$  intrări, care pot fi de mai multe feluri:

1. O fază de tăiere a copacilor, caracterizată de trei numere întregi  $l$ ,  $r$ , și  $k$ . În această fază, Azusa își va petrece următoarele  $k$  zile tăind copaci. În fiecare zi ea găsește cel mai înalt copac al cărei poziții este cuprinsă între  $l$  și  $r$  și îi scade acestuia înălțimea cu 1. În cazul în care există mai mulți astfel de copaci de înălțime maximă, ea îl alege pe cel mai din stânga. Dacă cel mai înalt copac are înălțimea 0, atunci nimic nu se întâmplă în acea zi.
2. O fază magică, caracterizată de două numere întregi  $i$  și  $x$ . În această fază, Azusa modifică copacul de pe poziția  $i$ , astfel încât acesta să aibă înălțimea  $x$ .
3. O fază de inspecție a copacilor, caracterizată de două numere întregi  $l$  și  $r$ . În această fază, Azusa va găsi suma înălțimilor copacilor ce au pozițiile cuprinse între  $l$  și  $r$ .

(Observați că termenul "cuprinse" înseamnă inclusiv capetele; de exemplu, 1, 2, 3, 4, 5 sunt "cuprinse" între 1 și 5.)

Azusa este curioasă care vor fi rezultatele fazelor de inspecție a copacilor și vrea să le știe fără să fie nevoită să parcurgă întreg orarul de una singură. Puteți să o ajutați voi?

## Protocol de Interacțiune

Concurentul trebuie să implementeze următoarele patru funcții:

```
void initialise(int N, int Q, int h[]);  
void cut(int l, int r, int k);  
void magic(int i, int x);  
long long int inspect(int l, int r);
```

Funcției `initialise` îi sunt date  $N$  (numărul copacilor),  $Q$  (numărul intrărilor din orar), și un vector  $h$ , unde  $h[i]$  este înălțimea celui de-al  $i$  copac, pentru  $1 \leq i \leq N$ . Această funcție este apelată de către sursa comisiei exact o singură dată, înainte ca oricare dintre celelalte trei funcții să fie apelate. Funcțiile `cut`, `magic` și `inspect` reprezintă fazele de tăiere a copacilor, magică și respectiv de inspecție a copacilor, și sunt caracterizate de respectivii lor parametri. Implementarea concurentului a funcției `inspect` trebuie să returneze suma înălțimilor copacilor cu poziții între  $l$  și  $r$ .

Concurentul nu trebuie să implementeze funcția `main`. Aceasta va fi implementată în fișierul comisiei `grader.cpp`; veți primi un exemplu `grader.cpp` în atașamente. Funcția noastră `main` va citi  $N$ ,  $Q$ , secvența celor  $N$  înălțimi inițiale, și cele  $Q$  intrări din orar. Cele trei tipuri de intrări din orar (`cut(l, r, k)`, `magic(i, x)` și `inspect(l, r)`) sunt codificate ca 1 l r k, 2 i x și respectiv 3 l r. Acesta este formatul fișierului de intrare ce va fi folosit în exemplele de mai jos.

Observați că îi este permis concurentului să folosească variabile globale, funcții suplimentare, metode și clase.



## Restricții

- $1 \leq N, Q \leq 300\,000$
- Se garantează faptul că funcțiile `cut`, `magic` și `inspect` vor fi apelate de exact  $Q$  ori în total.
- $1 \leq i \leq N$
- $0 \leq x, k, h[i] \leq 1\,000\,000\,000$
- $1 \leq l \leq r \leq N$

#	Punctaj	Restricții
1	5	$N \leq 1\,000, Q \leq 1\,000, k = 1$
2	8	$N \leq 80\,000, Q \leq 80\,000, k = 1$
3	8	$N \leq 1\,000, Q \leq 1\,000$ , nu există faze <code>magice</code> .
4	19	Nu există faze <code>magice</code> .
5	10	$l = 1, r = N$
6	21	$N \leq 80\,000, Q \leq 80\,000$
7	29	Nu există alte restricții suplimentare.

## Exemple

Intrare	Ieșire
6 10	9
1 2 3 1 2 3	6
1 1 6 3	5
3 1 6	1005
1 1 3 3	4
3 1 6	
1 1 3 1000	
3 1 6	
2 1 1000	
3 1 6	
1 1 3 999	
3 1 5	

## Explicație

În prima fază, după fiecare dintre cele 3 zile de tăiere de copaci, înălțimile copacilor sunt 1, 2, 2, 1, 2, 3; 1, 2, 2, 1, 2, 2; și 1, 1, 2, 1, 2, 2. Suma acestor valori este 9, care este răspunsul la inspecția din a doua fază.

În a treia fază, după fiecare dintre cele 3 zile de tăiere de copaci, înălțimile copacilor sunt 1, 1, 1, 1, 2, 2; 0, 1, 1, 1, 2, 2; și 0, 0, 1, 1, 2, 2. Suma acestor valori este 6, care este răspunsul la inspecția din a patra fază.

În a cincea fază, după fiecare din cele 1000 de zile de tăiere de copaci, înălțimile copacilor sunt 0, 0, 0, 1, 2, 2. Aceasta deoarece un copac cu înălțime 0 nu poate fi tăiat. Suma acestor valori este 5, care este răspunsul la inspecția din faza a șasea.

În a șaptea fază, primul copac este crescut la înălțimea 1000, rezultând în înălțimile 1000, 0, 0, 1, 2, 2. Suma acestor valori este 1005, care este răspunsul la inspecția din a opta fază.

În a noua fază, fiecare dintre cele 999 de zile de tăiere de copaci reduce înălțimea primului copac cu 1. Asta ne dă următoarele înălțimi de copaci 1, 0, 0, 1, 2, 2 la sfârșitul fazei. Suma primelor cinci valori dintre acestea este 4, care este răspunsul la inspecția din a zecea fază — faza finală.