



Feladat: Weirdtree

C++ header weirdtree.h

Azusa, a felföldi boszorkány, egy furábbnál furább fákkal teli kertbe botlott. A barátjával, Laikával, együtt úgy döntöttek, hogy gondozásba veszik a kertet.

A kert N fát tartalmaz, egy sorban. Ezeket (balról jobbra) 1-től N -ig számozzuk. Minden fának egy (nem-negatív) egész számmal írhatjuk le a magasságát. Azusa egy Q lépcsős tervet fog követni, ahol minden művelet a következők egyike:

1. Fametszés: ezt három egész szám (l , r és k) írja le. Ebben a lépésben Azusa k napig fákat metsz. Minden nap megkeresi a legmagasabb, l és r között elhelyezkedő fát és 1 egységgel csökkenti a magasságát. Ha több azonos magasságú legmagasabb fa van, akkor a legbalrább lévőét metszi meg. Ha a legmagasabb fa magassága 0, akkor semmi sem történik aznap.
2. Varázslat: ezt két egész szám (i és x) írja le. Azusa elvarázsolja az i . fát úgy, hogy az x magas legyen.
3. Mérés: ezt két egész szám (l és r) írja le. Azusa kiszámítja az l és r közötti fák magasságainak összegét.

(Az l és r közötti fák halmaza tartalmazza l -t és r -t, pl. 1 és 5 között ezek a fák vannak: 1, 2, 3, 4, 5.)

Azusa kíváncsi, hogy mi lesz a mérések eredménye, és anélkül szeretné azt megtudni, hogy végre kelljen hajtania az egész tervet. Segítesz neki?

Interakciós protokoll

Az alábbi 4 függvényt kódold le:

```
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);
```

A bizottság kódja először az `initialise` függvényt hívja meg, pontosan egyszer. A bemenet N (a fák száma), Q (a lépések száma) és egy h tömb, ahol $h[i]$ az i . fa magassága ($1 \leq i \leq N$). A `cut` (metszés), `magic` (varázslat) and `inspect` (mérés) a felfebb leírt műveleteket jelentik, ezeket a fenti paraméterekkel sorban fogja a bizottság meghívni. Az `inspect` függvény adja vissza a mérés eredményét, vagyis az l és r közötti fák magasságainak összegét.

A beadott kód ne tartalmazzon `main` függvényt, ezt majd a bizottság `grader.cpp` fájlja tartalmazza. A csatolmányok között letölthető egy minta `grader.cpp`, amit tesztelésre használhatsz. A mintakód először N és Q értékét olvassa be a standard input-ról, ezt követi az N eredeti magasságérték, majd a Q lépés. A három fajta műveletet (`cut(l, r, k)`, `magic(i, x)`, `inspect(l, r)`) rendre 1 1 r k, 2 i x és 3 l r formában kódoljuk. A lenti példákban is ezt a kódolást használjuk.

Természetesen kedved szerint használhatsz globális változókat, saját függvényeket és osztályokat a kódodban.



Korlátok

- $1 \leq N, Q \leq 300\,000$
- A `cut`, `magic` and `inspect` függvényeket összesen pontosan Q alkalommal fogjuk meghívni.
- $1 \leq i \leq N$
- $0 \leq x, k, h[i] \leq 1\,000\,000\,000$
- $1 \leq l \leq r \leq N$

#	Pontszám	Korlátok
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	$N \leq 1\,000, Q \leq 1\,000$, és nincsenek <code>magic</code> (varázslat) műveletek.
4	19	Nincsenek <code>magic</code> (varázslat) műveletek.
5	10	$l = 1, r = N$
6	21	$N \leq 80\,000, Q \leq 80\,000$
7	29	Nincsenek további korlátok

Példák

Bemenet	Kimenet
6 10 1 2 3 1 2 3 1 1 6 3 3 1 6 1 1 3 3 3 1 6 1 1 3 1000 3 1 6 2 1 1000 3 1 6 1 1 3 999 3 1 5	9 6 5 1005 4

Magyarázat

Az első lépésben Azusa 3 napig metsz, ezalatt a magasságok naponta a következők: 1, 2, 2, 1, 2, 3; 1, 2, 2, 1, 2, 2; és 1, 1, 2, 1, 2, 2. Az utolsó számsor összege, így a második lépésben a mérés eredménye, 9.

A harmadik lépésben a magasságok naponta: 1, 1, 1, 1, 2, 2; 0, 1, 1, 1, 2, 2; és 0, 0, 1, 1, 2, 2. Az utolsó számsor összege 6, így ez lesz a negyedik lépésben a mérés eredménye.

Az ötödik lépésben mind az ezer nap után 0, 0, 0, 1, 2, 2 lesz a fák magassága, mivel a 0 magasságú fák nem metszhetők. Az összeg 5, így a hatodik lépésben a mérésnél is ezt kapjuk eredményül.

A hetedik lépésben az első fát 1000 magassá varázsolja Azusa, ezután a fák magassága rendre 1000, 0, 0, 1, 2, 2. Az összeg 1005, ez a válasz a nyolcadik lépésben.

A kilencedik lépésben mind a 999 napon eggyel csökken az első fa magassága, így a fák magassága végül rendre 1, 0, 0, 1, 2, 2. Az első 5 fa magasságának összege 4, így a tizedik lépésben a mérés eredménye is 4 lesz.