



## Aufgabe Weirdtree

C++ header      `weirdtree.h`

Hochlandhexe Azusa hat auf wundersame Weise einen Garten voller bizarrer Bäume entdeckt! Zusammen mit ihrer Freundin Laika hat sie sich entschieden, sich für einige Zeit niederzulassen und den Garten zu hegen.

Der Garten kann als Folge von  $N$  Bäumen betrachtet werden, in der die Bäume von 1 bis  $N$  durchnummeriert sind. Jeder Baum hat eine bestimmte Höhe, repräsentiert als nichtnegative Ganzzahl. Azusa plant ihre Gartenarbeit gemäß einem Zeitplan aus  $Q$  Einträgen. Diese entsprechen jeweils einem der folgenden Typen:

1. **Schneiden:** Gegeben sind drei Ganzzahlen  $l$ ,  $r$  und  $k$ . Azusa verbringt die nächsten  $k$  Tage damit, Bäume zu schneiden. Jeden Tag findet sie den höchsten Baum, dessen Index zwischen  $l$  und  $r$  liegt, und reduziert die Höhe um 1. Gibt es mehrere höchste Bäume, nimmt sie denjenigen, der sich am weitesten links befindet. Hat der höchste Baum Höhe 0, passiert nichts an diesem Tag.
2. **Magie:** Gegeben sind zwei Ganzzahlen  $i$  und  $x$ . Azusa verzaubert den Baum an Index  $i$ ; dieser hat nun Höhe  $x$ .
3. **Inspizieren:** Gegeben sind zwei Ganzzahlen  $l$  und  $r$ . Azusa möchte die Summe der Höhen der Bäume mit Indizes zwischen  $l$  und  $r$  berechnen.

(Bemerke, dass hier beim Wort “zwischen” der Bereich immer inklusiv angegeben ist; z.B. liegen die Werte 1, 2, 3, 4, 5 alle “zwischen” 1 und 5.)

Azusa möchte jetzt schon wissen, was die Resultate der Inspektionen sein werden. Kannst du ihr helfen?

## Interaktives Protokoll

Implementiere die folgenden vier Funktionen:

```
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);
```

Die Funktion `initialise` gibt dir  $N$  (die Anzahl Bäume),  $Q$  (die Anzahl Einträge im Zeitplan) und einem Array  $h$ , wobei  $h[i]$  die Höhe des  $i$ -ten Baums angibt, für  $1 \leq i \leq N$ . Diese Funktion wird vom Grader genau einmal aufgerufen, vor den anderen drei Funktionen. Die Funktionen `cut`, `magic` und `inspect` geben die Einträge des Zeitplanes an für Schneiden, Magie und Inspizieren, mit den jeweils entsprechenden Parametern. Deine Implementierung der Funktion `inspect` soll die Summe der Höhen der Bäume zwischen  $l$  und  $r$  zurückgeben.

Die `main`-Funktion wird vom Grader in `grader.cpp` bereitgestellt; du findest sie im `grader.cpp` im Anhang zur Aufgabe. Der Beispiel-Grader liest  $N$ ,  $Q$ , dann eine Liste der  $N$  ursprüngliche Höhen, und die  $Q$  Einträge im Zeitplan. Die drei Typen (`cut(l, r, k)`, `magic(i, x)` bzw. `inspect(l, r)`) sind angegeben mit 1 1 r k, 2 i x bzw. 3 l r. Dies ist auch das Eingabeformat, das in den Beispielen unten benutzt wird.

Es ist dir erlaubt, globale Variable, zusätzliche Funktionen, Methoden und/oder Klassen zu benutzen.



## Limits

- $1 \leq N, Q \leq 300\,000$
- Es ist garantiert, dass `cut`, `magic` und `inspect` insgesamt genau  $Q$  Mal aufgerufen werden.
- $1 \leq i \leq N$
- $0 \leq x, k, h[i] \leq 1\,000\,000\,000$
- $1 \leq l \leq r \leq N$

#	Punkte	Limits
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	$N \leq 1\,000, Q \leq 1\,000$ , und es hat keine <code>magic</code> -Operation.
4	19	Es gibt keine <code>magic</code> -Operation.
5	10	$l = 1, r = N$
6	21	$N \leq 80\,000, Q \leq 80\,000$
7	29	Keine weiteren Beschränkungen

## Beispiele

Eingabedatei	Ausgabedatei
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

## Bemerkungen

Nach der ersten Operation, nach 3 Tagen Bäume schneiden, sind die Höhen der Bäume 1, 2, 2, 1, 2, 3; 1, 2, 2, 1, 2, 2; und 1, 1, 2, 1, 2, 2. Die Summe dieser Werte ist 9, und dies ist die Antwort auf die Inspektion der zweiten Operation.

In der dritten Operation, nach 3 Tagen Bäume schneiden, sind die Höhen der Bäume 1, 1, 1, 1, 2, 2; 0, 1, 1, 1, 2, 2; und 0, 0, 1, 1, 2, 2. Die Summe dieser Werte ist 6, und dies ist die Antwort auf die Inspektion in der vierten Operation.

In der fünften Operation, nach 100 Tagen Bäume schneiden, sind die Höhen der Bäume 0, 0, 0, 1, 2, 2. Dies, weil Bäume von Höhe 0 nicht kleiner geschnitten werden können. Die Summe dieser Werte ist 5, und dies ist die Antwort auf die Inspektion in der sechsten Operation.

In der siebten Operation wächst der erste Baum auf Höhe 1000 und die Höhen der Bäume sind danach 1000, 0, 0, 1, 2, 2. Die Summe all dieser Werte ist 1005, und dies ist die Antwort auf die Inspektion in der achten Operation.

In der neunten Operation, nach 999 Tagen Bäume schneiden, hat sich die Höhe des ersten Baums auf 1 geändert. Dies gibt uns die Höhen 1, 0, 0, 1, 2, 2 am Ende der Operation. Die Summe der ersten fünf dieser Werte ist 4, und dies ist die Antwort auf die Inspektion in der zehnten und letzten Operation.