

사탕 분배

앤티 콩은 가까운 학교의 학생들을 위해 n 개의 사탕 박스를 준비하고 있다. 박스는 0 부터 $n - 1$ 로 나타내고, 초기에는 비어있다. 박스 i ($0 \leq i \leq n - 1$)는 최대 $c[i]$ 개(박스의 용량)의 사탕을 담을 수 있다.

앤티 콩은 q 일동안 박스들을 준비한다. j ($0 \leq j \leq q - 1$)번째 날에, 그녀는 세 개 정수들 $l[j]$, $r[j]$, $v[j]$ 로 나타내는 행동을 수행한다. 여기서, $0 \leq l[j] \leq r[j] \leq n - 1$ 이고 $v[j] \neq 0$ 이다.

$l[j] \leq k \leq r[j]$ 를 만족하는 각 박스 k 에 대해서 :

- $v[j] > 0$ 이면, 앤티 콩은 정확히 $v[j]$ 개의 사탕을 넣었거나 또는 박스가 가득 찰 때까지 박스 k 에 사탕을 하나씩 넣는다. 다시 말해서, 행동 전에 박스에 p 개의 사탕이 있었다면, 행동 후에 박스는 $\min(c[k], p + v[j])$ 개의 사탕을 가지게 된다.
- $v[j] < 0$ 이면, 앤티 콩은 정확히 $-v[j]$ 개의 사탕을 제거했거나 또는 박스가 빌 때까지 박스 k 에서 사탕을 하나씩 제거한다. 다시 말해서, 행동 전에 박스에 p 개의 사탕이 있었다면, 행동 후에 박스는 $\max(0, p + v[j])$ 개의 사탕을 가지게 된다.

당신이 해야 할 일은 q 일이 지난 후, 각 박스의 사탕 개수를 결정하는 것이다.

Implementation Details

당신은 다음 프로시저를 구현해야 한다:

```
int[] distribute_candies(int[] c, int[] l, int[] r, int[] v)
```

- c : 길이 n 의 배열. $0 \leq i \leq n - 1$ 에 대해서, $c[i]$ 는 박스 i 의 용량이다.
- l , r , v : 길이 q 의 세 배열. j ($0 \leq j \leq q - 1$) 번째 날에 앤티 콩은 위에서 말한대로 정수들 $l[j]$, $r[j]$, $v[j]$ 가 나타내는 행동을 수행한다.
- 이 프로시저는 길이 n 의 배열을 반환해야 한다. 그 배열을 s 로 나타내면, $0 \leq i \leq n - 1$ 에 대해서, $s[i]$ 는 q 일 후에 박스 i 의 사탕 개수여야 한다.

Examples

Example 1

다음 호출을 생각한다:

```
distribute_candies([10, 15, 13], [0, 0], [2, 1], [20, -11])
```

박스 0은 사탕 10개의 용량, 박스 1은 사탕 15개의 용량, 박스 2는 사탕 13개의 용량을 가짐을 의미한다.

0 일의 끝에, 박스 0은 $\min(c[0], 0 + v[0]) = 10$ 개의 사탕, 박스 1은 $\min(c[1], 0 + v[0]) = 15$ 개의 사탕, 박스 2는 $\min(c[2], 0 + v[0]) = 13$ 개의 사탕을 가진다.

1 일의 끝에, 박스 0은 $\max(0, 10 + v[1]) = 0$ 개의 사탕, 박스 1은 $\max(0, 15 + v[1]) = 4$ 개의 사탕을 가진다. $2 > r[1]$ 때문에 박스 2의 사탕 개수에는 변화가 없다. 각 요일의 끝에 사탕 개수는 아래와 같다:

Day	Box 0	Box 1	Box 2
0	10	15	13
1	0	4	13

프로시저는 $[0, 4, 13]$ 을 반환해야 한다.

Constraints

- $1 \leq n \leq 200\,000$
- $1 \leq q \leq 200\,000$
- $1 \leq c[i] \leq 10^9$ ($0 \leq i \leq n - 1$)
- $0 \leq l[j] \leq r[j] \leq n - 1$ ($0 \leq j \leq q - 1$)
- $-10^9 \leq v[j] \leq 10^9, v[j] \neq 0$ ($0 \leq j \leq q - 1$)

Subtasks

1. (3 points) $n, q \leq 2000$
2. (8 points) $v[j] > 0$ ($0 \leq j \leq q - 1$)
3. (27 points) $c[0] = c[1] = \dots = c[n - 1]$
4. (29 points) $l[j] = 0$ 그리고 $r[j] = n - 1$ ($0 \leq j \leq q - 1$)
5. (33 points) 추가 제약조건 없음.

Sample Grader

샘플 그레이더는 다음 형식으로 입력을 읽는다:

- line 1: n
- line 2: $c[0] c[1] \dots c[n - 1]$
- line 3: q
- line 4 + j ($0 \leq j \leq q - 1$): $l[j] r[j] v[j]$

샘플 그레이더는 다음 형식으로 답을 출력한다:

- line 1: $s[0] s[1] \dots s[n - 1]$