Repartiendo caramelos

La tía Khong está preparando n cajas de caramelos para los estudiantes de una escuela cercana. Las cajas se numeran desde 0 hasta $n-1$ y se encuentran inicialmente vacías. La caja i ($0 \leq i \leq n-1$) tiene una capacidad de $c[i]$ caramelos.

La tía Khong pasa q días preparando las cajas. En el día j ($0 \leq j \leq q-1$), realiza una acción especificada mediante tres enteros $l[j]$, $r[j]$ y $v[j]$ donde $0 \leq l[j] \leq r[j] \leq n-1$ y $v[j] \neq 0$. Para cada caja k tal que $l[j] \leq k \leq r[j]$:

- Si $v[j] > 0$, la tía Khong agrega caramelos a la caja k, uno por uno, hasta que haya agregado exactamente $v[j]$ caramelos o la caja se haya llenado. En otras palabras, si la caja tenía p caramelos antes de realizar la acción, tendrá $\min(c[k], p + v[j])$ caramelos luego de llevarla a cabo.
- Si $v[j] < 0$, la tía Khong quita caramelos de la caja k, uno por uno, hasta que haya quitado exactamente $-v[j]$ caramelos o la caja se haya vaciado. En otras palabras, si la caja tenía p caramelos antes de realizar la acción, tendrá $\max(0, p + v[j])$ caramelos luego de llevarla a cabo.

Tu tarea consiste en determinar el número de caramelos en cada caja, luego de los q días.

Detalles de implementación

Debes implementar la siguiente función:

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

- c: un arreglo de longitud n. Para $0 \leq i \leq n-1$, $c[i]$ indica la capacidad de la caja i.
- l, r y v: tres arreglos de longitud q. En el día j, para $0 \leq j \leq q-1$, la tía Khong realiza una acción especificada mediante los enteros $l[j]$, $r[j]$ y $v[j]$, como se describió más arriba.
- Esta función debe retornar un arreglo de longitud n. Llamamos s a este arreglo. Para $0 \leq i \leq n-1$, $s[i]$ debe ser la cantidad de caramelos en la caja i luego de los q días.

Ejemplos

Ejemplo 1

Considera la siguiente llamada:

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

Esto significa que la caja 0 tiene una capacidad de 10 caramelos, la caja 1 tiene una capacidad de 15 caramelos, y la caja 2 tiene una capacidad de 13 caramelos.

Al terminar el día 0, la caja 0 tiene $\min(c[0], 0 + v[0]) = 10$ caramelos, la caja 1 tiene $\min(c[1], 0 + v[0]) = 15$ caramelos y la caja 2 tiene $\min(c[2], 0 + v[0]) = 13$ caramelos.

Al terminar el día 1, la caja 0 tiene $\max(0, 10 + v[1]) = 0$ caramelos, la caja 1 tiene $\max(0, 15 + v[1]) = 4$ caramelos. Como $2 > r[1]$, la cantidad de caramelos en la caja 2 no cambia. La cantidad de caramelos al finalizar cada día se resume a continuación:

Día	Caja 0	Caja 1	Caja 2
0	10	15	13
1	0	4	13

Por lo tanto, la función debe retornar $[0, 4, 13]$.

Restricciones

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

Subtareas

1. (3 puntos) $n, q \leq 2000$
2. (8 puntos) $v[j] > 0$ (para todo $0 \leq j \leq q - 1$)
3. (27 puntos) $c[0] = c[1] = \ldots = c[n - 1]$
4. (29 puntos) $l[j] = 0$ y $r[j] = n - 1$ (para todo $0 \leq j \leq q - 1$)
5. (33 puntos) Sin más restricción.

Evaluador Local

El evaluador local lee la entrada con el siguiente formato:

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

El evaluador local escribe tu respuesta con el siguiente formato:

- línea 1: $s[0]$ $s[1]$ $\ldots$ $s[n-1]$