

Distributing Candies

Aunty Khong esta preparando n cajas de caramelos para estudiantes de una escuela cercana. Las cajas están numeradas desde 0 a $n - 1$ e inicialmente están vacías. La caja i ($0 \leq i \leq n - 1$) tiene una capacidad de $c[i]$ caramelos.

Aunty Khong pasa q días preparando las cajas. En el día j ($0 \leq j \leq q - 1$), ella hace una acción descrita por tres enteros $l[j]$, $r[j]$ y $v[j]$ donde $0 \leq l[j] \leq r[j] \leq n - 1$ y $v[j] \neq 0$. Por cada caja k que satisface $l[j] \leq k \leq r[j]$:

- Si $v[j] > 0$, Aunty Khong agrega caramelos a la caja k , uno por uno, hasta que haya agregado exactamente $v[j]$ caramelos o hasta que la caja se llene. En otras palabras, si la caja tenía p caramelos antes de la acción, tendrá $\min(c[k], p + v[j])$ caramelos después de la acción.
- Si $v[j] < 0$, Aunty Khong saca caramelos de la caja k , uno por uno, hasta que haya sacado exactamente $-v[j]$ caramelos o hasta que la caja se vacíe. En otras palabras, si la caja tiene p caramelos antes de la acción, tendrá $\max(0, p + v[j])$ caramelos después de la acción.

Tu tarea es determinar el número de caramelos en cada caja después de q días.

Detalles de Implementación

Debes implementar el siguiente procedimiento:

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

- c : un arreglo de longitud n . Para $0 \leq i \leq n - 1$, $c[i]$ denota la capacidad de la caja i
- l , r y v : tres arreglos de longitud q . En el día j , para $0 \leq j \leq q - 1$, Aunty Khong realiza una acción descrita por los enteros $l[j]$, $r[j]$ y $v[j]$, como se ha explicado arriba.
- Este procedimiento debe retornar un arreglo de longitud n . Llamemoslo arreglo s . Para $0 \leq i \leq n - 1$, $s[i]$ debe ser el número de caramelos en la caja i después de 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 final del 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 final del 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 $2 > r[1]$, no hay cambio en el número de caramelos en la caja 2. El número de caramelos al final de cada día está sumariados abajo:

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

Entonces, el procedimiento 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) $c[0] = c[1] = \dots = c[n - 1]$
4. (29 puntos) $l[j] = 0$ y $r[j] = n - 1$ (para todo $0 \leq j \leq q - 1$)
5. (33 puntos) No hay restricciones adicionales.

Evaluador de ejemplo

El evaluador de ejemplo lee la entrada en el siguiente formato:

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

El evaluador de ejemplo imprime tus respuestas en el siguiente formato t:

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