

Distribuyendo dulces

La Tía Khong está preparando n cajas de dulces para estudiantes de una escuela cercana. Las cajas son numeradas de 0 a $n - 1$ y están inicialmente vacías. La caja i ($0 \leq i \leq n - 1$) tiene una capacidad de $c[i]$ dulces.

La Tía Khong se lleva q días para preparar las cajas. En el día j ($0 \leq j \leq q - 1$), ella realiza la acción especificada por tres enteros $l[j]$, $r[j]$ y $v[j]$, donde $0 \leq l[j] \leq r[j] \leq n - 1$ y $v[j] \neq 0$. Para cada caja k que satisface $l[j] \leq k \leq r[j]$:

- Si $v[j] > 0$, la Tía Khong agrega dulces a la caja k , uno por uno, hasta que ha agregado exactamente $v[j]$ dulces o hasta que la caja esté llena. En otras palabras, si la caja tenía p dulces antes de la acción, tendrá $\min(c[k], p + v[j])$ dulces después de la acción.
- Si $v[j] < 0$, la Tía Khong quita dulces de la caja k , uno por uno, hasta que ha removido exactamente $-v[j]$ dulces o hasta que la caja esté vacía. En otras palabras, si la caja tenía p dulces antes de la acción, tendrá $\max(0, p + v[j])$ después de la acción.

Tu tarea será determinar el número de dulces en cada caja después de q días.

Detalles de Implementación

Deberás implementar la siguiente función:

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

- c : un arreglo de tamaño n . Para $0 \leq i \leq n - 1$, $c[i]$ denota la capacidad de la caja i .
- l , r y v : tres arreglos de tamaño q . En el día j , para $0 \leq j \leq q - 1$, la Tía Khong realiza una acción especificada por los enteros $l[j]$, $r[j]$ y $v[j]$, como se describió anteriormente.
- Esta función deberá retornar un arreglo de tamaño n . Se denota dicho arreglo por s . Para $0 \leq i \leq n - 1$, $s[i]$ deberá ser el número de dulces en la caja i después de los q días.

Ejemplos

Ejemplo 1

Considera la llamada siguiente:

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

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

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

Al final del día 1, la caja 0 tiene $\max(0, 10 + v[1]) = 0$ dulces y la caja 1 tiene $\max(0, 15 + v[1]) = 4$ dulces. Ya que $2 > r[1]$, no hay cambio en el número de dulces de la caja 2. El número de dulces al final de cada día se resume en la siguiente tabla:

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

Es decir, la función deberá devolver $[0, 4, 13]$.

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) Sin restricciones adicionales.

Calificador de Ejemplo

El calificador 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 calificador de ejemplo imprime tus respuestas en el siguiente formato:

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