

Роздавання смаколиків

Онті Конг готує n коробок смаколиків для учнів найближчої школи. Коробки пронумеровано від 0 до $n - 1$, на початку вони порожні. Коробка i ($0 \leq i \leq n - 1$) може вміщати $c[i]$ смаколиків.

Онті Конг готує коробки протягом q днів. У день j ($0 \leq j \leq q - 1$) вона виконує дію, що задається трьома цілими числами $l[j]$, $r[j]$ та $v[j]$, де $0 \leq l[j] \leq r[j] \leq n - 1$ та $v[j] \neq 0$. Для кожної коробки k відомо, що $l[j] \leq k \leq r[j]$:

- Якщо $v[j] > 0$, Онті Конг додає смаколики у коробку k , один за одним, поки вона не додасть рівно $v[j]$ смаколиків або поки коробка не заповниться. Іншими словами, якщо у коробці було p смаколиків, то після виконання цієї дії стане $\min(c[k], p + v[j])$.
- Якщо $v[j] < 0$, Онті Конг забирає смаколики з коробки k , один за одним, поки не забере рівно $-v[j]$ смаколиків або поки коробка не стане порожньою. Іншими словами, якщо у коробці було p смаколиків, то після виконання цієї дії стане $\max(0, p + v[j])$.

Ви маєте визначити кількість смаколиків у кожній коробці після q днів.

Деталі реалізації

Ви маєте реалізувати наступну процедуру:

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

- c : масив довжини n . Для $0 \leq i \leq n - 1$, $c[i]$ задає місткість коробки i .
- l , r та v : три масиви довжини q . У день j , при $0 \leq j \leq q - 1$, Онті Конг виконує дію, задану числами $l[j]$, $r[j]$ and $v[j]$, як описано вище.
- Ця процедура має повернути масив довжини n . Назвемо його s . Для $0 \leq i \leq n - 1$, $s[i]$ має бути кількістю смаколиків у коробці i після q днів.

Приклади

Приклад 1

Дозглянемо наступний виклик:

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

Це означає, що коробка 0 містить 10 смаколиків, коробка 1 містить 15 смаколиків, а коробка 2 містить 13 смаколиків.

Наприкінці дня 0: коробка 0 містить $\min(c[0], 0 + v[0]) = 10$ смаколиків, коробка 1 містить $\min(c[1], 0 + v[0]) = 15$ смаколиків, а коробка 2 містить $\min(c[2], 0 + v[0]) = 13$ смаколиків.

Наприкінці дня 1: коробка 0 містить $\max(0, 10 + v[1]) = 0$ смаколиків, коробка 1 містить $\max(0, 15 + v[1]) = 4$ смаколиків. Оскільки $2 > r[1]$, кількість смаколиків у коробці 2 не змінюється. Кількість смаколиків наприкінці кожного дня наведено у таблиці нижче:

День	Коробка 0	Коробка 1	Коробка 2
0	10	15	13
1	0	4	13

Процедура має повернути $[0, 4, 13]$.

Обмеження

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

Підзадачі

1. (3 бали) $n, q \leq 2000$
2. (8 балів) $v[j] > 0$ (для всіх $0 \leq j \leq q - 1$)
3. (27 балів) $c[0] = c[1] = \dots = c[n - 1]$
4. (29 балів) $l[j] = 0$ та $r[j] = n - 1$ (для всіх $0 \leq j \leq q - 1$)
5. (33 бали) без додаткових обмежень.

Приклад модуля перевірки

Приклад модуля перевірки читає вхідні дані у наступному форматі:

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

Приклад модуля перевірки друкує ваші відповіді у наступному форматі:

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