Rozdzielanie cukierków (Distributing Candies)

Cioteczka Khong przygotowuje n pudełek z cukierkami dla uczniów z pobliskiej szkoły. Pudełka są ponumerowane liczbami od 0 do $n-1$ oraz początkowo są puste. Pudełko i ($0 \le i \le n-1$) może pomieścić do $c[i]$ cukierków.

Cioteczka Khong poświęca q dni na przygotowanie pudełek. W dniu j ($0 \le j \le q-1$) wykonuje akcję opisaną trzema liczbami całkowitymi $l[j]$, $r[j]$ i $v[j]$, gdzie $0 \le l[j] \le r[j] \le n-1$ oraz $v[j] \neq 0$. Dla każdego pudełka k spełniającego $l[j] \le k \le r[j]$:

- Jeśli $v[j] > 0$, cioteczka Khong dodaje cukierki do pudełka k, jeden po drugim, dopóki nie doda dokładnie $v[j]$ cukierków lub pudełko stanie się pełne. Inaczej mówiąc, jeśli w pudełku przed wykonaniem akcji było p cukierków, to po wykonaniu akcji będzie w nim $\min(c[k], p + v[j])$ cukierków.
- Jeśli $v[j] < 0$, cioteczka Khong wyjmuje cukierki z pudełka k, jeden po drugim, dopóki nie wyjmie dokładnie $-v[j]$ cukierków lub pudełko stanie się puste. Inaczej mówiąc, jeśli w pudełku przed wykonaniem akcji było p cukierków, to po wykonaniu akcji będzie w nim $\max(0, p + v[j])$ cukierków.

Twoim zadaniem jest wyznaczenie liczby cukierków w każdym pudełku po q dniach.

Szczegóły implementacyjne

Powinieneś zaimplementować następującą funkcję:

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

- c: tablica długości n. Dla $0 \le i \le n-1$, $c[i]$ oznacza pojemność pudełka i.
- l, r i v: trzy tablice długości q. W dniu j, dla $0 \le j \le q-1$, cioteczka Khong wykonuje akcję opisaną liczbami całkowitymi $l[j]$, $r[j]$ oraz $v[j]$.
- Wynikiem działania funkcji powinna być tablica długości n. Oznaczmy tę tablicę przez s. Dla $0 \le i \le n-1$, $s[i]$ powinno być równe liczbie cukierków w pudełku i po q dniach.

Przykłady

Przykład 1

Rozważmy następujące wywołanie:

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

To oznacza, że pudełko 0 może pomieścić 10 cukierków, pudełko 1 może pomieścić 15 cukierków, a pudełko 2 może pomieścić 13 cukierków.

Na koniec dnia 0, pudełko 0 zawiera $\min(c[0], 0 + v[0]) = 10$ cukierków, pudełko 1 zawiera $\min(c[1], 0 + v[0]) = 15$ cukierków, a pudełko 2 zawiera $\min(c[2], 0 + v[0]) = 13$ cukierków.

Na koniec dnia 1, pudełko 0 zawiera $\max(0, 10 + v[1]) = 0$ cukierków, a pudełko 1 zawiera $\max(0, 15 + v[1]) = 4$ cukierków. Ponieważ $2 > r[1]$, liczba cukierków w pudełku 2 nie ulega zmianie. Poniższa tabelka przedstawia liczby cukierków w pudełkach pod koniec każdego dnia:

Dzień	Pudełko 0	Pudełko 1	Pudełko 2
0	10	15	13
1	0	4	13

Wynikiem działania funkcji powinno więc być $[0, 4, 13]$.

Ograniczenia

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

Podzadania

1. (3 punkty) $n, q \leq 2000$
2. (8 punktów) $v[j] > 0$ (dla każdego $0 \leq j \leq q - 1$)
3. (27 punktów) $c[0] = c[1] = \ldots = c[n - 1]$
4. (29 punktów) $l[j] = 0$ oraz $r[j] = n - 1$ (dla każdego $0 \leq j \leq q - 1$)
5. (33 punktów) Brak dodatkowych ograniczeń.

Przykładowa sprawdzaczka

Przykładowa sprawdzaczka wczytuje dane wejściowe w następującym formacie:

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

Przykładowa sprawdzaczka wypisuje Twoje odpowiedzi w następującym formacie:

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