

# Distributing Candies

Дејче спрема  $n$  кутии со слатки за посетителите на Доручек на Камени мост. Кутиите се нумерирани со  $0$  до  $n - 1$  и на почеток се празни. Кутијата  $i$  ( $0 \leq i \leq n - 1$ ) има капацитет од  $c[i]$  слатки.

Дејче троши  $q$  денови за подготовка на кутиите. На  $j$ -иот ден ( $0 \leq j \leq q - 1$ ), тој ја прави следната акција зададена преку 3 цели броја  $l[j]$ ,  $r[j]$  и  $v[j]$  каде  $0 \leq l[j] \leq r[j] \leq n - 1$  и  $v[j] \neq 0$ . За секоја кутија  $k$  за која  $l[j] \leq k \leq r[j]$ :

- Ако  $v[j] > 0$ , Дејче додава слатки во кутијата  $k$ , една по една, се дури не додаде  $v[j]$  слатки или дури не ја наполни кутијата. Поинаку кажано, ако во кутијата имало  $p$  слатки пред акцијата, ќе има  $\min(c[k], p + v[j])$  слатки по акцијата.
- Ако  $v[j] < 0$ , Дејче вади слатки во кутијата  $k$ , една по една, се дури не извади  $v[j]$  слатки или дури не ја испразни кутијата. Поинаку кажано, ако во кутијата имало  $p$  слатки пред акцијата, ќе има  $\max(0, p + v[j])$  слатки по акцијата. Ваша задача е да го одредите бројот на слатки во секоја кутија после  $q$ -те денови.

## Implementation Details

Треба да ја имплементирате процедурата:

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

- $c$ : низа со должина  $n$ . За  $0 \leq i \leq n - 1$ ,  $c[i]$  го дава капацитетот на кутијата  $i$ .
- $l$ ,  $r$  и  $v$ : три низи со должина  $q$ . На денот  $j$ , за  $0 \leq j \leq q - 1$ , Дејче прави акција зададена со броевите  $l[j]$ ,  $r[j]$  и  $v[j]$ , според погорниот опис.
- Оваа процедура треба да врати низа со должина  $n$ . Означете ја низата со  $s$ . За  $0 \leq i \leq n - 1$ ,  $s[i]$  треба да е бројот на слатки во кутијата  $i$  после  $q$  дена.

## Examples

### Example 1

Нека е даден следниот повик:

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

Ова значи дека кутијата  $0$  има капацитет од  $10$  слатки, кутијата  $1$  капацитет од  $15$  слатки, а кутијата  $2$  капацитет од  $13$  слатки.

На крај на денот 0, кутијата 0 има  $\min(c[0], 0 + v[0]) = 10$  слатки, кутијата 1 има  $\min(c[1], 0 + v[0]) = 15$  слатки и кутијата 2 има  $\min(c[2], 0 + v[0]) = 13$  слатки.

На крај на денот 1, кутијата 0 има  $\max(0, 10 + v[1]) = 0$  слатки, кутијата 1 има  $\max(0, 15 + v[1]) = 4$  слатки. Бидејќи  $2 > r[1]$ , нема промена во бројот на слатки во кутијата 2. Бројот на слатки после секоја акција е даден подолу во табелата:

Ден	Кутија 0	Кутија 1	Кутија 2
0	10	15	13
1	0	4	13

Така, процедурата треба да врати  $[0, 4, 13]$ .

## Constraints

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

## Subtasks

1. (3 поени)  $n, q \leq 2000$
2. (8 поени)  $v[j] > 0$  (за сите  $0 \leq j \leq q - 1$ )
3. (27 поени)  $c[0] = c[1] = \dots = c[n - 1]$
4. (29 поени)  $l[j] = 0$  и  $r[j] = n - 1$  (за сите  $0 \leq j \leq q - 1$ )
5. (33 поени) Без дополнителни ограничувања.

## Sample Grader

Sample grader-от го чита влезот во следниот формат:

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

Sample grader-от го печати одговорот во следниот формат:

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