

Распределение конфет

Тетя Хонг готовит n коробок конфет для учеников ближайшей школы. Коробки пронумерованы от 0 до $n - 1$ и изначально пустые. Коробка номер i ($0 \leq i \leq n - 1$) может вместить в себя $c[i]$ конфет.

Тетя Хонг потратит q дней на приготовление коробок. На j -й день ($0 \leq j \leq q - 1$), ее действия определяются тремя числами $l[j]$, $r[j]$ и $v[j]$, где $0 \leq l[j] \leq r[j] \leq n - 1$ и $v[j] \neq 0$. А именно, для каждой коробки с номером k , где $l[j] \leq k \leq r[j]$, происходит следующее:

- Если $v[j] > 0$, Тетя Хонг добавляет в коробку номер k конфеты по одной до тех пор, пока она не добавит ровно $v[j]$ конфет или пока коробка не станет полностью заполненной. Другими словами, если коробка содержала p конфет, то после этой операции она будет содержать $\min(c[k], p + v[j])$ конфет.
- Если $v[j] < 0$, Тетя Хонг убирает конфеты из коробки номер k по одной до тех пор, пока она не уберет ровно $-v[j]$ конфет или пока коробка не станет пустой. Другими словами, если коробка содержала p конфет, то после этой операции она будет содержать $\max(0, p + v[j])$ конфет.

Необходимо определить количество конфет в каждой из коробок после q дней.

Детали реализации

Вам необходимо реализовать следующие функции:

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

- c : массив длины n . Для всех $0 \leq i \leq n - 1$ число $c[i]$ означает вместимость коробки с номером i .
- l , r и v : три массива длины q . На день j , для всех $0 \leq j \leq q - 1$, действия Тети Хонг определяются числами $l[j]$, $r[j]$ и $v[j]$, согласно описанию выше.
- Функция должна вернуть массив длины n . Обозначим его за s . Для каждого $0 \leq i \leq n - 1$ число $s[i]$ должно быть равно количеству конфет в коробке номер i после q дней.

Примеры

Пример 1

Рассмотрим следующий вызов функции:

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

В этом примере коробка 0 имеет вместимость 10 конфет, коробка 1 имеет вместимость 15 конфет, а коробка 2 имеет вместимость 13 конфет.

В конце дня 0 в коробке 0 находится $\min(c[0], 0 + v[0]) = 10$ конфет, в коробке 1 — $\min(c[1], 0 + v[0]) = 15$ конфет, а в коробке 2 — $\min(c[2], 0 + v[0]) = 13$ конфет.

В конце дня 1 в коробке 0 находится $\max(0, 10 + v[1]) = 0$ конфет, в коробке 1 — $\max(0, 15 + v[1]) = 4$ конфеты. Так как $2 > r[1]$, количество конфет в коробке номер 2 не изменится. Количество конфет в каждой коробке в конце каждого дня представлено в таблице ниже:

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

Таким образом, функция должна вернуть $[0, 4, 13]$.

Ограничения

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

Подзадачи

1. (3 балла) $n, q \leq 2000$
2. (8 баллов) $v[j] > 0$ (для всех $0 \leq j \leq q - 1$)
3. (27 баллов) $c[0] = c[1] = \dots = c[n - 1]$
4. (29 баллов) $l[j] = 0$ и $r[j] = n - 1$ (для всех $0 \leq j \leq q - 1$)
5. (33 балла) Без дополнительных ограничений.

Пример грейдера

Грейдер считывает данные в следующем формате:

- строка 1: n
- строка 2: $c[0] \ c[1] \ \dots \ c[n - 1]$
- строка 3: q

- строка $4 + j$ ($0 \leq j \leq q - 1$): $l[j] \ r[j] \ v[j]$

Грейдер выводит результат вызова функции в следующем формате:

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