

Membagikan Permen

Bibi Khong sedang menyiapkan n buah kotak berisi permen untuk dibagikan ke murid dari sekolah terdekat. Kotak-kotak tersebut dinomori dari 0 sampai $n - 1$ dan awalnya kosong. Kotak i ($0 \leq i \leq n - 1$) memiliki kapasitas sebesar $c[i]$ permen.

Bibi Khong menghabiskan q hari untuk menyiapkan kotak-kotak tersebut. Pada hari ke- j ($0 \leq j \leq q - 1$), dia melakukan sebuah aksi yang dinyatakan dengan tiga buah bilangan bulat $l[j]$, $r[j]$ dan $v[j]$ dimana $0 \leq l[j] \leq r[j] \leq n - 1$ dan $v[j] \neq 0$. Untuk setiap kotak k yang memenuhi $l[j] \leq k \leq r[j]$:

- Apabila $v[j] > 0$, Bibi Khong menambahkan beberapa permen ke kotak k , satu per satu, sampai dia telah menambahkan tepat $v[j]$ permen atau sampai kotak tersebut penuh. Dengan kata lain, apabila kotak tersebut memiliki p permen sebelumnya, kotak itu akan memiliki $\min(c[k], p + v[j])$ permen setelahnya.
- Apabila $v[j] < 0$, Bibi Khong membuang beberapa permen dari kotak k , satu per satu, sampai dia telah membuang tepat $-v[j]$ permen atau sampai kotak tersebut kosong. Dengan kata lain, apabila kotak tersebut memiliki p permen sebelumnya, kotak itu akan memiliki $\max(0, p + v[j])$ permen setelahnya.

Anda bertugas untuk menentukan banyaknya permen di tiap kotak setelah q hari.

Detail Implementasi

Anda harus mengimplementasikan fungsi berikut:

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

- c : sebuah array berukuran n . Untuk $0 \leq i \leq n - 1$, $c[i]$ menyatakan kapasitas dari kotak i .
- l , r dan v : Tiga buah array berukuran q . Pada hari ke- j , untuk $0 \leq j \leq q - 1$, Bibi Khong melakukan sebuah aksi yang dinyatakan dengan tiga buah bilangan bulat $l[j]$, $r[j]$ dan $v[j]$, seperti yang telah dideskripsikan diatas.
- Fungsi ini harus mengembalikan sebuah array berukuran n . Sebut array tersebut dengan s . Untuk $0 \leq i \leq n - 1$, $s[i]$ harus merupakan banyaknya permen di kotak i setelah q hari.

Contoh

Contoh 1

Perhatikan pemanggilan berikut:

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

Hal ini berarti kotak 0 memiliki kapasitas 10 permen, kotak 1 memiliki kapasitas 15, dan kotak 2 memiliki kapasitas 13 permen.

Diakhir dari hari 0, kotak 0 memiliki $\min(c[0], 0 + v[0]) = 10$ permen, kotak 1 memiliki $\min(c[1], 0 + v[0]) = 15$ permen dan kotak 2 memiliki $\min(c[2], 0 + v[0]) = 13$ permen.

Pada akhir hari 1, kotak 0 memiliki $\max(0, 10 + v[1]) = 0$ permen, kotak 1 memiliki $\max(0, 15 + v[1]) = 4$ permen. Karena $2 > r[1]$, tidak ada perubahan dalam banyaknya permen pada kotak 2. Banyaknya permen pada tiap harinya dapat diringkas sebagai berikut:

Hari	Kotak 0	Kotak 1	Kotak 2
0	10	15	13
1	0	4	13

Oleh karena itu, fungsi ini harus mengembalikan $[0, 4, 13]$.

Batasan

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

Subsoal

1. (3 poin) $n, q \leq 2000$
2. (8 poin) $v[j] > 0$ (untuk semua $0 \leq j \leq q - 1$)
3. (27 poin) $c[0] = c[1] = \dots = c[n - 1]$
4. (29 poin) $l[j] = 0$ dan $r[j] = n - 1$ (untuk semua $0 \leq j \leq q - 1$)
5. (33 poin) Tidak ada batasan tambahan.

Contoh Grader

Contoh *grader* membaca input dengan format sebagai berikut:

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

Contoh *grader* mencetak jawaban Anda dengan format sebagai berikut:

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