

Distributing Candies

Mătușa Khong pregătește n cutii de bomboane pentru elevii de la o școală din apropiere. Cutiile sunt numerotate de la 0 la $n - 1$ și sunt inițial goale. În cutia i ($0 \leq i \leq n - 1$) încap $c[i]$ bomboane.

Mătușei Khong îi ia q zile să pregătească cutiile cu bomboane. În ziua j ($0 \leq j \leq q - 1$), ea efectuează o acțiune specificată de trei numere întregi $l[j]$, $r[j]$ și $v[j]$ unde

$0 \leq l[j] \leq r[j] \leq n - 1$ și $v[j] \neq 0$. Pentru fiecare cutie k , astfel încât $l[j] \leq k \leq r[j]$:

- Dacă $v[j] > 0$, atunci mătușa Khong adaugă bomboane în cutia k , una câte una, până când sunt adăugate exact $v[j]$ bomboane sau cutia devine plină. Cu alte cuvinte, dacă în cutie erau p bomboane înainte de acțiune, atunci după acțiune în cutie vor fi $\min(c[k], p + v[j])$ bomboane.
- Dacă $v[j] < 0$, atunci mătușa Khong scoate bomboanele din cutia k , una câte una, până când scoate exact $-v[j]$ bomboane sau cutia devine goală. Cu alte cuvinte, dacă cutia avea p bomboane înainte de acțiune, atunci după acțiune în cutie vor fi $\max(0, p + v[j])$ bomboane.

Sarcina dvs. este de a determina numărul de bomboane din fiecare cutie după q zile.

Detalii de implementare

Trebuie să implementați următoarea procedură:

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

- c : un tablou unidimensional de lungime n . Pentru $0 \leq i \leq n - 1$, $c[i]$ reprezintă numărul de bomboane care încap în cutia i .
- l , r și v : trei tablouri unidimensionale de lungime q . În ziua j , pentru $0 \leq j \leq q - 1$, mătușa Khong efectuează o acțiune specificată de numerele întregi $l[j]$, $r[j]$ și $v[j]$, așa cum este descris mai sus.
- Această procedură ar trebui să returneze un tablou unidimensional de lungime n . Notăm tabloul cu s . Pentru $0 \leq i \leq n - 1$, $s[i]$ ar trebui să fie numărul bomboanelor din cutia i după q zile.

Exemplu

Să considerăm următorul apel:

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

Aceasta înseamnă că în cutia 0 încap 10 bomboane, în cutia 1 încap 15 bomboane, iar în cutia 2 încap 13 bomboane.

La sfârșitul zilei 0, cutia 0 are $\min(c[0], 0 + v[0]) = 10$ bomboane, cutia 1 are $\min(c[1], 0 + v[0]) = 15$ bomboane și cutia 2 are $\min(c[2], 0 + v[0]) = 13$ bomboane.

La sfârșitul zilei 1, cutia 0 are $\max(0, 10 + v[1]) = 0$ bomboane, cutia 1 are $\max(0, 15 + v[1]) = 4$ bomboane. Deoarece $2 > r[1]$, nu există nicio modificare a numărului de bomboane din cutia 2. Numărul de bomboane la sfârșitul fiecărei zile este rezumat mai jos:

| Ziua | Cutia 0 | Cutia 1 | Cutia 2 |
|------|---------|---------|---------|
| 0 | 10 | 15 | 13 |
| 1 | 0 | 4 | 13 |

Prin urmare procedura ar trebui să returneze $[0, 4, 13]$.

Restricții

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

Subtask-uri

1. (3 puncte) $n, q \leq 2000$
2. (8 puncte) $v[j] > 0$ (pentru oricare $0 \leq j \leq q - 1$)
3. (27 puncte) $c[0] = c[1] = \dots = c[n - 1]$
4. (29 puncte) $l[j] = 0$ și $r[j] = n - 1$ (pentru oricare $0 \leq j \leq q - 1$)
5. (33 puncte) Fără restricții suplimentare.

Exemplul de Grader

Grader-ul citește datele de intrare în următorul format:

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

Grader-ul afișează valoarea returnată în următorul format:

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