

# Cukríky (Distributing Candies)

Tetka Betka chystá deťom  $n$  balíčkov s cukríkmi. Balíčky majú čísla od  $0$  po  $n - 1$ . Na začiatku sú všetky prázdne. Balíček  $i$  ( $0 \leq i \leq n - 1$ ) má kapacitu  $c[i]$  cukríkov.

Príprava balíčkov trvá  $q$  dní. Každý deň spraví tetka Betka jednu akciu. Presnejšie, v deň  $j$  ( $0 \leq j \leq q - 1$ ) spraví akciu, ktorú popisujú tri celé čísla  $l[j]$ ,  $r[j]$  a  $v[j]$ . Pre tieto čísla platí  $0 \leq l[j] \leq r[j] \leq n - 1$  a  $v[j] \neq 0$ . Akcia sa bude týkať tých balíčkov  $k$ , ktoré spĺňajú  $l[j] \leq k \leq r[j]$ . S každým z nich Betka spraví nasledovné:

- Ak  $v[j] > 0$ , tetka Betka bude postupne po jednom pridávať cukríky do balíčku  $k$ , až kým buď nepridá  $v[j]$  cukríkov alebo balíček  $k$  nenaplní. Inými slovami, ak bolo pred akciou v tomto balíčku  $p$  cukríkov, po akcii tam bude  $\min(c[k], p + v[j])$  cukríkov.
- Ak  $v[j] < 0$ , tetka Betka bude postupne po jednom odoberať cukríky z balíčku  $k$ , až kým buď neodoberie  $-v[j]$  cukríkov alebo balíček  $k$  nevyprázdni. Inými slovami, ak bolo pred akciou v tomto balíčku  $p$  cukríkov, po akcii tam bude  $\max(0, p + v[j])$  cukríkov.

Zisti, koľko bude nakoniec v ktorom balíčku cukríkov.

## Detaily implementácie

Tvojou úlohou je implementovať nasledovnú funkciu:

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

- $c$ : pole dĺžky  $n$ . Pre  $0 \leq i \leq n - 1$ ,  $c[i]$  označuje maximálny počet cukríkov v balíčku  $i$ .
- $l$ ,  $r$  and  $v$ : tri polia dĺžky  $q$ . V deň  $j$ , pre  $0 \leq j \leq q - 1$ , Betka spraví akciu, ktorú vyššie uvedeným spôsobom popisujú  $l[j]$ ,  $r[j]$  a  $v[j]$ .
- Návratovou hodnotou je pole dĺžky  $n$ . Označme ho  $s$ . Pre  $0 \leq i \leq n - 1$ ,  $s[i]$  má byť počet cukríkov, ktoré skončia v balíčku  $i$  po tom, ako Betka postupne vykoná všetkých  $q$  akcií.

## Príklady

### Príklad 1

Uvažujme toto volanie tvojej funkcie:

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

Zadané parametre nám hovoria, že balíček 0 má kapacitu 10 cukríkov, balíček 1 má kapacitu 15 cukríkov a do balíčka 2 sa zmestí nanajvýš 13 cukríkov.

Po dni 0 bude v balíčku 0 presne  $\min(c[0], 0 + v[0]) = 10$  cukríkov, v balíčku 1 ich bude  $\min(c[1], 0 + v[0]) = 15$  a v balíčku 2 sa bude nachádzať  $\min(c[2], 0 + v[0]) = 13$  cukríkov.

Po dni 1 bude v balíčku 0 presne  $\max(0, 10 + v[1]) = 0$  candies a v balíčku 1 bude  $\max(0, 15 + v[1]) = 4$  cukríkov. Keďže balíček 2 neležal v úseku, ktorý Betka tento deň menila ( $r[1] < 2$ ), počet cukríkov v tomto balíčku sa nemení.

To isté ešte raz ako tabuľka:

po dni	balíček 0	balíček 1	balíček 2
0	10	15	13
1	0	4	13

Tvoja funkcia má vrátiť počty cukríkov po poslednom dni, teda pole  $[0, 4, 13]$ .

## Obmedzenia

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

## Podúlohy

1. (3 points)  $n, q \leq 2000$
2. (8 points)  $v[j] > 0$  (pre všetky  $0 \leq j \leq q - 1$ )
3. (27 points)  $c[0] = c[1] = \dots = c[n - 1]$
4. (29 points)  $l[j] = 0$  a  $r[j] = n - 1$  (pre všetky  $0 \leq j \leq q - 1$ )
5. (33 points) bez ďalších obmedzení

## Ukážkový grader

Ukážkový grader očakáva vstup v nasledovnom formáte:

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

Ukážkový grader vypíše návratovú hodnotu tvojej funkcie v nasledujúcom formáte:

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