

Wild operations (wild)

Filipo želi da testira Frančeskovu sposobnost u baratanju ludim operacijama sa nizovima, pa mu je dao niz A_0, \dots, A_{N-1} dužine N .

Sada će Filippo tražiti od Frančeska da izvrši neke operacije na nizu, gde svaka operacija može biti:

- *promeni* vrednost A_p u x , za neki ceo broj x i validan indeks p .
- *poremeti* opseg $[l, r]$, tj. da postavi $A_p = \max(A_p, A_{p-1})$ **istovremeno** za sve $l < p \leq r$.

U bilo kom trenutku, Filippo može tražiti od Frančeska da mu kaže vrednost A_p za neki validan indeks p .

Frančesko je veoma zauzet, pa je odlučio da zatraži tvoju pomoć da odgovori na Filipova pitanja.

Имплементација

Moraš predati jedan fajl sa ekstenzijom `.cpp`.

↩ Među priložima za ovaj zadatak pronaći ćeš šablon `wild.cpp` sa primerom implementacije.

Moraš implementirati sledeće funkcije:

C++	<code>void init(int N, vector<int> A);</code>
-----	---

- Ova funkcija se poziva jednom, na početku izvršavanja tvog programa.
- Ceo broj N je dužina niza.
- Niz A , indeksiran od 0 do $N - 1$, je početni niz koji je Filippo izabrao.

C++	<code>void change(int p, int x);</code>
-----	---

- Ova funkcija se poziva mnogo puta tokom izvršavanja tvog programa, kada Filippo izvrši promenu.
- Ceo broj p je indeks promenjene vrednosti u nizu.
- Ceo broj x je nova vrednost koja se dodeljuje.

C++	<code>void perturb(int l, int r);</code>
-----	--

- Ova funkcija se poziva mnogo puta tokom izvršavanja tvog programa, kada Filippo poremeti opseg.
- Ceo broj l je levi kraj opsega koji Filippo poremećuje.
- Ceo broj r je desni kraj opsega koji Filippo poremećuje.

C++	<code>int calc(int p);</code>
-----	-------------------------------

- Ova funkcija se poziva mnogo puta tokom izvršavanja tvog programa, kada Filippo pita za vrednost elementa niza.
- Ceo broj p je indeks elementa za koji je Filippo pitao.
- Funkcija treba da vrati vrednost A_p nakon primene svih prethodnih operacija.

Primer Grejdera

Pojednostavljena verzija grejdera korišćenog tokom ispravke dostupna je u direktorijumu vezanom za ovaj problem. Možeš ga koristiti za lokalno testiranje svojih rešenja. Primer grejder čita ulazne podatke iz `stdin`, poziva funkcije koje treba da implementiraš i piše u `stdout` u sledećem formatu.

Neka je Q ukupan broj promena, poremećaja i pitanja koje je Filipino uradio.

Zatim, ulazni fajl se sastoji od $2 + Q$ linija, koje sadrže:

- Linija 1: celi brojevi N, Q .
- Linija 2: N celih brojeva A_0, \dots, A_{N-1} , početne vrednosti niza.
- Linija $3 + i$ ($0 \leq i < Q$): 2 ili 3 cela broja, u jednom od sledećih formata:
 - $1\ p\ x$: što znači da Filipino menja A_p u x .
 - $2\ l\ r$: što znači da Filipino poremećuje opseg $[l, r]$;
 - $3\ p$: što znači da Filipino pita za vrednost A_p .

Izlazni fajl se sastoji od Q_3 linija (gde je Q_3 broj poziva funkcije `calc`) koje sadrže vrednosti koje vraća funkcija `calc`.

Ограничења

- $1 \leq N \leq 400\,000$.
- $0 \leq Q \leq 400\,000$.
- $1 \leq A_i \leq 10^9$ za sve $0 \leq i < N$.
- $0 \leq p < N$ pri svakom pozivu `change` i `calc`.
- $0 \leq l < r \leq N - 1$ pri svakom pozivu `perturb`.
- $1 \leq x \leq 10^9$ pri svakom pozivu `change`.

Бодовање

Tvoj program će biti testiran na više test primera grupisanih u podzadatke. Da bi dobio poene za podzadatak, moraš rešiti sve test primere u njemu ispravno.

Neka je Q_1 broj poziva funkcije `change` u test primeru, tada:

- **Подзатак 0 [0 поена]:** Primer.
- **Подзатак 1 [15 поена]:** Funkcija `change` nikada nije pozvana; $l = 0, r = N - 1$ pri svakom pozivu `perturb`.
- **Подзатак 2 [16 поена]:** $A_i \leq 10$ za sve $0 \leq i < N$ i $x \leq 10$ za sve pozive `change`.
- **Подзатак 3 [13 поена]:** Pozivi funkcije `change` ne smanjuju vrednosti ($x \geq A_p$), $Q_1 \leq 1000$ i $l = 0, r = N - 1$ pri svakom pozivu `perturb`.
- **Подзатак 4 [22 поена]:** Funkcija `change` nikada nije pozvana.
- **Подзатак 5 [14 поена]:** Pozivi funkcije `change` ne smanjuju vrednosti ($x \geq A_p$), $Q_1 \leq 1000$.
- **Подзатак 6 [20 поена]:** Nema dodatnih ograničenja.

Примери улаза/излаза

stdin	stdout
10 28	1
5 1 7 8 3 2 5 6 9 4	3
1 1 1	1
1 0 1	7
2 0 1	8
2 2 6	1
1 6 5	8
2 2 9	3
2 2 5	6
2 4 5	4
1 4 5	9
2 3 8	
1 8 4	
3 0	
1 6 3	
1 4 1	
2 5 7	
1 0 3	
2 4 5	
1 6 3	
3 0	
3 1	
3 2	
3 3	
3 4	
3 5	
3 6	
3 7	
3 8	
3 9	

Објашњење

Počinjemo sa nizom $A = [5, 1, 7, 8, 3, 2, 5, 6, 9, 4]$.

- Događaj 1: Filipo menja A_1 u 1 (već je bio 1): novi niz je $[5, 1, 7, 8, 3, 2, 5, 6, 9, 4]$.
- Događaj 2: Filipo menja A_0 u 1: novi niz je $[1, 1, 7, 8, 3, 2, 5, 6, 9, 4]$.
- Događaj 3: Filipo poremećuje $[0, 1]$: novi niz je $[1, 1, 7, 8, 3, 2, 5, 6, 9, 4]$.
- Događaj 4: Filipo poremećuje $[2, 6]$: novi niz je $[1, 1, 7, 8, 8, 3, 5, 6, 9, 4]$.

Od događaja 19 pa nadalje, Filipo samo пита за вредности u nizu bez vršenja promena ili poremećaja. U ovom trenutku niz je $[3, 1, 7, 8, 1, 8, 3, 6, 4, 9]$.