

Pimpketova čarobna trava

Vleposednik Pimpke voli da gaji travu, baš kao što su to činili i njegovi preci. On sada poseduje najveći travnjak u Crnotravskom okrugu. No, to nije oduvek bio slučaj. Pre N godina, Pimpke je bio običan lerd (srpska reč za *onaj koji nosi travu*) i imao je svega jedno jutro trave. Sanjao je o tome da zaradi mnogo para i konačno postane baja (srpska reč za *čovek sa zlatnom kajlom*).

Označimo godine od 0 do $N - 1$, hronološki (tj. godina $N - 1$ je najskorija). Svake godine su tržišni uslovi uticali na porast Pimpketovog travnjaka. Za svaku godinu i , Pimpke zna pozitivan ceo broj $X[i]$. Ako je godinu i počeo sa h jutara (h je ceo broj), tu godinu je završio sa travnjakom veličine $h \cdot X[i]$ jutara.

Na kraju svake godine Pimpke je mogao prodati određeni broj jutara (ceo broj). Za svaku godinu i , Pimpke zna pozitivan ceo broj $Y[i]$: cenu za koju je mogao prodati jutro na kraju te godine. Nakon svake godine mogao je prodati proizvoljan (ceo) broj jutara, svako po ceni $Y[i]$.

Pimpke se pita koliko bi najviše para do sada mogao zaraditi da je odabrao najbolje momente da proda svoja jutra u toku prethodnih N godina. Imali ste čast da vas Pimpke pozove na slavu (srpska reč za *prekomerno uživanje u različitim porocima*), i tom prilikom je od vas zatražio da odgovorite na njegovo pitanje.

Dok dejstvo poroka slabi, Pimpketovo sećanje postaje sve bolje, i on pravi M izmena u podacima koje je naveo. Svaka izmena menja ili jednu od $X[i]$ vrednosti ili jednu od $Y[i]$ vrednosti. Nakon svake izmene on vas iznova pita koliko je najviše para mogao da zaradi prodavajući svoja jutra. Pimpketove izmene su kumulativne: svaki vaš odgovor treba da uzme u obzir sve prethodne izmene. Svako $X[i]$ ili $Y[i]$ može biti izmenjeno više puta.

Tačan odgovor na Pimpketovo pitanje može biti ogroman. Da biste izbegli rad sa velikim brojevima, od vas se samo traži da saopštite odgovor po modulu $10^9 + 7$.

Primer

Neka je $N = 3$, sa sledećim vrednostima za X i Y :

	0	1	2
X	2	1	3
Y	3	4	1

Za ove početne vrednosti, Pimpke može zaraditi najviše ako proda oba svoja jutra trave na kraju godine 1. Ceo proces bi izgledao ovako:

- Na početku Pimpke ima 1 jutro.

- Nakon godine 0 će imati $1 \cdot X[0] = 2$ jutra.
- Nakon godine 1 će imati $2 \cdot X[1] = 2$ jutra.
- On sada može prodati ta 2 jutra. Ukupna zarada će biti $2 \cdot Y[1] = 8$.

Zatim, pretpostavimo da imamo $M = 1$ izmenu: menjamo $Y[1]$ u 2.

Nakon ove izmene X i Y su:

	0	1	2
X	2	1	3
Y	3	2	1

U ovom slučaju, jedno od optimalnih rešenja je da Pimpke proda jedno jutro trave na kraju godine 0 a zatim 3 jutra na kraju godine 2. Ceo proces bi izgledao ovako:

- Na početku Pimpke ima 1 jutro.
- Nakon godine 0 imaće $1 \cdot X[0] = 2$ jutra.
- On sada može prodati jedno od njih za $Y[0] = 3$, i preostaće mu 1 jutro.
- Nakon godine 1 imaće $1 \cdot X[1] = 1$ jutro.
- Nakon godine 2 imaće $1 \cdot X[2] = 3$ jutra.
- On sada može prodati ta 3 jutra za $3 \cdot Y[2] = 3$, što čini ukupnu zaradu od $3 + 3 = 6$.

Zadatak

Date su vam vrednosti N , X , Y , i lista izmena. Pre prve izmene, i nakon svake izmene, izračunajte najveću moguću zaradu koju Pimpke može imati od prodaje svojih jutara trave, po modulu $10^9 + 7$. Treba da implementirate funkcije `init`, `updateX` i `updateY`.

- `init(N, X, Y)` — Grejder će pozvati prvo ovu funkciju i pozvaće je tačno jednom.
 - N : ukupan broj godina.
 - X : niz dužine N . Za $0 \leq i \leq N - 1$, $X[i]$ predstavlja koliko puta se umnožio broj jutara u godini i .
 - Y : niz dužine N . Za $0 \leq i \leq N - 1$, $Y[i]$ predstavlja cenu jutra na kraju godine i .
 - Primetite da X i Y predstavljaju početne Pimpketove vrednosti (pre bilo kakvih izmena).
 - Nakon što se funkcija `init` završi, nizovi X i Y ostaju važeći i možete menjati vrednosti u njima po želji.
 - Funkcija treba da vrati vrednost koja predstavlja najveću moguću zaradu koju Pimpke može da ostvari za početne vrednosti nizova X i Y , po modulu $10^9 + 7$.
- `updateX(pos, val)`
 - `pos`: ceo broj iz intervala $0, \dots, N - 1$.

- `val`: nova vrednost za $X[\text{pos}]$.
- Funkcija treba da vrati vrednost koja predstavlja najveću moguću zaradu koju Pimpke može da ostvari nakon ove izmene, po modulu $10^9 + 7$.
- `updateY(pos, val)`
 - `pos`: ceo broj iz intervala $0, \dots, N - 1$.
 - `val`: nova vrednost za $Y[\text{pos}]$.
 - Funkcija treba da vrati vrednost koja predstavlja najveću moguću zaradu koju Pimpke može da ostvari nakon ove izmene, po modulu $10^9 + 7$.

Možete pretpostaviti da su sve početne vrednosti, kao i izmenjene vrednosti $X[i]$ i $Y[i]$ između 1 i 10^9 , uključivo.

Nakon pozivanja funkcije `init`, grejder će pozivati funkcije `updateX` i `updateY` više puta. Ukupan broj poziva funkcija `updateX` i `updateY` je M .

Podzadaci

podzadatak	bodovi	N	M	dodatna ograničenja
1	17	$1 \leq N \leq 10$	$M = 0$	$X[i], Y[i] \leq 10$, $X[0] \cdot X[1] \cdot \dots \cdot X[N - 1] \leq 1,000$
2	17	$1 \leq N \leq 1,000$	$0 \leq M \leq 1,000$	nema
3	20	$1 \leq N \leq 500,000$	$0 \leq M \leq 100,000$	$X[i] \geq 2$ i $val \geq 2$ za funkcije <code>init</code> i <code>updateX</code> respektivno
4	23	$1 \leq N \leq 500,000$	$0 \leq M \leq 10,000$	nema
5	23	$1 \leq N \leq 500,000$	$0 \leq M \leq 100,000$	nema

Primer grejdera

Grejder čita podatke iz fajla `horses.in` u sledećem formatu:

- linija 1: N
- linija 2: $X[0] \dots X[N - 1]$
- linija 3: $Y[0] \dots Y[N - 1]$
- linija 4: M
- linije 5, ..., $M + 4$: tri broja `type pos val` (`type=1` za `updateX` i `type=2` za `updateY`).

Grejder prvo štampa vrednost koju vraća funkcija `init` a zatim i povratne vrednosti funkcija `updateX` i `updateY` u redosledu u kome se pozivaju.