Krzysztof Diks, Lech Duraj
Tłumaczenie

Dostępna pamięć: 1500 MB, limit czasu: 1,5 s. *IOI 2015, dzień drugi, 30.07.2015*

Konie

Mansur, wzorem swoich starożytnych przodków, jest zapalonym hodowcą koni. Posiada obecnie największe stado w całym Kazachstanie. Nie zawsze jednak tak było. Ongiś, N lat temu, Mansur był jedynie **dżygitem** *(kaz.* **młodym człowiekiem***) mającym tylko jednego konia. Marzył wtedy, aby zarobić dużo pieniędzy i stać się* **bajem** *(kaz.* **bogaczem***).*

Ponumerujmy lata działalności Mansura liczbami $0, 1, \ldots, N-1$, od najdawniejszych do najnowszych. Podczas każdego roku stado mogło zwiększyć swoją liczebność, w zależności od pogody. Dla każdego roku i, Mansur pamięta współczynnik wzrostu – całkowitą dodatnią liczbę $X[i]$. Jeśli na początku roku i stado liczyło h koni, na koniec tego roku było już $h \cdot X[i]$ koni w stadzie.

Konie można było sprzedawać wyłącznie pod koniec roku. Dla każdego roku i, Mansur pamięta również dodatnią liczbę całkowitą $Y[i]$ – cenę, jaką można było otrzymać za sprzedaż jednego konia. Po każdym roku można było sprzedać dowolnie wiele spośród posiadanych koni, wszystkie po tej samej cenie $Y[i]$.

Mansur zastanawia się, ile najwięcej pieniędzy mógł był zarobić, gdyby wybrał najlepsze momenty do sprzedaży koni w ciągu tych N lat. Masz obecnie zaszczyt gościć u Mansura w czasie **toi** *(kaz.* **wakacji***), Mansur zadał więc to pytanie Tobie.*

Dodatkowo, w miarę snucia opowieści, Mansur przypomina sobie nowe fakty. Dokona on zatem M poprawek swojej historii. Każda z poprawek zmieni dokładnie jedną wartość $X[i]$ albo dokładnie jedną wartość $Y[i]$. Po każdej poprawce Mansur znowu zapyta Cię o największą sumę pieniędzy, jaką mógł zarobić. Poprawki Mansura kumulują się – każda z Twoich odpowiedzi musi brać pod uwagę wszystkie poprzednie poprawki. Zwróć też uwagę na fakt, że każde $X[i]$ lub $Y[i]$ może być zmieniane wielokrotnie.

Odpowiedzi na pytania Mansura mogą być wielkimi sumami pieniędzy. Aby uniknąć dużych liczb, podaj wszystkie odpowiedzi modulo $10^9 + 7$.

Przykład

Załóżmy, że historia trwa $N = 3$ lata z następującymi parametrami:

rok	0	1	2
X	2	1	3
Y	3	4	1

Dla tych wartości Mansur zarobi najwięcej, jeśli sprzeda oba swoje konie pod koniec roku 1. Cała historia wygląda wtedy następująco:

- *Na początku Mansur posiada 1 konia.*
- *Po roku 0 będzie miał $1 \cdot X[0] = 2$ konie.*
- *Po roku 1 będzie miał $2 \cdot X[1] = 2$ konie.*
- *Teraz może sprzedać oba konie za łączną sumę $2 \cdot Y[1] = 8$.*

Teraz załóżmy, że jest $M = 1$ poprawka: zmiana $Y[1]$ na 2. Po poprawce dane są następujące:

rok	*0*	*1*	*2*
X	*2*	*1*	*3*
Y	*3*	*2*	*1*

W tym wypadku jednym z możliwych najlepszych rozwiązań jest sprzedaż jednego konia po roku 0, a trzech po roku 2. Cała historia wygląda wtedy następująco:

- *Na początku Mansur posiada 1 konia.*
- *Po roku 0 będzie miał $1 \cdot X[0] = 2$ konie.*
- *Sprzedaje jednego z koni po cenie $Y[0] = 3$, pozostaje mu więc jeden koń.*
- *Po roku 1 będzie miał $1 \cdot X[1] = 1$ konia.*
- *Po roku 2 będzie miał $1 \cdot X[2] = 3$ konie.*
- *Teraz sprzedaje wszystkie konie za sumę $3 \cdot Y[2] = 3$. Zarobił łącznie $3 + 3 = 6$.*

Zadanie

Dane są liczba N, tablice X, Y oraz lista poprawek. Oblicz maksymalny zysk ze sprzedaży koni Mansura przed wszystkimi poprawkami, a także po każdej poprawce. Wyniki podaj modulo $10^9 + 7$.

Musisz zaimplementować funkcje `init`, `updateX` *i* `updateY`.

- `init(N, X, Y)` – *Program sprawdzający wywoła tę funkcję dokładnie raz.*
  - `N`*: liczba lat.*
  - `X`*: tablica długości* `N`*. Dla każdego $0 \leqslant i \leqslant$* `N` *$- 1$,* `X[i]` *to współczynnik wzrostu w roku i.*
  - `Y`*: tablica długości* `N`*. Dla każdego $0 \leqslant i \leqslant$* `N` *$- 1$,* `Y[i]` *to cena konia po roku i.*
  - *Wartości* `X` *oraz* `Y` *to liczby podane przez Mansura na początku (przed wszystkimi poprawkami).*
  - *Po zakończeniu działania funkcji* `init` *tablice* `X` *i* `Y` *pozostają zaalokowane i możesz dowolnie modyfikować ich zawartość.*
  - *Funkcja powinna zwrócić maksymalny zysk Mansura dla podanych początkowych wartości* `X` *oraz* `Y`*, modulo $10^9 + 7$.*
- `updateX(pos, val)`
  - `pos`*: liczba całkowita z zakresu $0, \ldots, N - 1$.*
  - `val`*: nowa wartość $X[$*`pos`*$]$.*
  - *Funkcja powinna zwracać maksymalny zysk Mansura po uwzględnieniu tej poprawki, modulo $10^9 + 7$.*

- `updateY(pos, val)`
  - `pos`*: liczba całkowita z zakresu* $0, \ldots, N-1$.
  - `val`*: nowa wartość* $Y[$`pos`$]$.
  - *Funkcja powinna zwracać maksymalny zysk Mansura po uwzględnieniu tej poprawki, modulo* $10^9 + 7$.

Możesz założyć, że zarówno początkowe, jak i zaktualizowane później wartości $X[i]$ *oraz* $Y[i]$ *są liczbami z zakresu od 1 do* 10^9 *włącznie. Po jednokrotnym wywołaniu* `init`*, program sprawdzający wywoła funkcje* `updateX` *oraz* `updateY` *pewną liczbę razy. Łączna liczba wywołań* `updateX` *i* `updateY` *będzie wynosiła* M.

Podzadania

podzadanie	**liczba punktów**	N	M	**dodatkowe ograniczenia**
1	17	$1 \leqslant N \leqslant 10$	$M = 0$	$X[i], Y[i] \leqslant 10$, $X[0] \cdot X[1] \cdot \ldots \cdot X[N-1] \leqslant 1000$
2	17	$1 \leqslant N \leqslant 1000$	$0 \leqslant M \leqslant 1000$	*brak*
3	20	$1 \leqslant N \leqslant 500\,000$	$0 \leqslant M \leqslant 100\,000$	$X[i] \geqslant 2$ *oraz* *val* $\geqslant 2$ *odpowiednio dla* `init` *i* `updateX`
4	23	$1 \leqslant N \leqslant 500\,000$	$0 \leqslant M \leqslant 10\,000$	*brak*
5	23	$1 \leqslant N \leqslant 500\,000$	$0 \leqslant M \leqslant 100\,000$	*brak*

Przykładowy program sprawdzający

Przykładowy program sprawdzający czyta dane z pliku `horses.in` *w następującej postaci:*

- *wiersz 1:* `N`
- *wiersz 2:* `X[0] ... X[N - 1]`
- *wiersz 3:* `Y[0] ... Y[N - 1]`
- *wiersz 4:* `M`
- *wiersze 5, …,* `M` *+ 4: trzy liczby* `type pos val` *(*`type` *= 1 dla* `updateX` *oraz* `type` *= 2 dla* `updateY`*).*

Przykładowy program sprawdzający wypisuje wartość zwróconą przez funkcję `init`*, a po niej wartości zwrócone przez wywołania funkcji* `updateX` *i* `updateY`*.*