

Caballos

Al igual que a sus ancestros, a Mansur le gusta mucho criar caballos. En este momento él posee la rebaño más grande en Kazajistán. Pero no siempre ha sido así. Hace N años, Mansur era simplemente un dzhiigit (*hombre joven* en kazajo) y únicamente tenía un caballo. El soñaba con hacer mucho dinero y convertirse finalmente en un bai (*una persona muy rica* en kazajo).

Numeremos los años de 0 a $N - 1$ en orden cronológico (esto es el año $N - 1$ es el más reciente). El clima de cada año influyó en el crecimiento del ganado. Para cada año i , Mansur se acuerda de un entero positivo, el coeficiente de crecimiento $X[i]$. Si usted comenzó el año i con h caballos, entonces usted terminó con $h \cdot X[i]$ caballos en su rebaño.

Los caballos se pueden vender únicamente al final de un año. Para cada año i , Mansur recuerda un entero positivo $Y[i]$: el precio por el cual él podría vender un caballo al final del año i . Después de cada año, era posible vender arbitrariamente muchos caballos, cada uno por el mismo precio $Y[i]$.

Mansur se pregunta cuál es la cantidad más grande que podría tener en el momento actual si él hubiera elegido el mejor momento para vender sus caballos durante los N años. Usted ha tenido el honor de ser invitado al toi (palabra kasaja para *vacaciones*) y él le pidió a usted que respondiera esta pregunta.

La memoria de Mansur se mejora a lo largo de la tarde, y entonces él hace una sucesión de M actualizaciones. Cada actualización cambiará uno de los valores $X[i]$ o uno de los valores $Y[i]$. Después de cada actualización, él le pide a usted nuevamente la mayor cantidad posible de dinero que él podría haber ganado vendiendo sus caballos. Las actualizaciones de Mansur son acumulativas: cada una de sus respuestas debería tener en cuenta todas las actualizaciones previas. Note que un mismo $X[i]$ o $Y[i]$ puede ser actualizado varias veces.

Las respuestas a las preguntas de Mansur pueden ser bastante grandes. Para evitar trabajar con números demasiado grandes se le pide que reporte la respuesta módulo $10^9 + 7$.

Ejemplo

Suponga que hay $N = 3$ años, con la siguiente información:

| | | | |
|---|---|---|---|
| | 0 | 1 | 2 |
| X | 2 | 1 | 3 |
| Y | 3 | 4 | 1 |

Para estos valores iniciales, Mansur puede ganar la mayor cantidad posible si él vende sus caballos al final del año 1. Todo el proceso se verá como sigue:

- Inicialmente, Mansur tiene 1 caballo.

- Después del año 0, él tendrá $1 \cdot X[0] = 2$ caballos.
- Después del año 1 el tendrá $2 \cdot X[1] = 2$ caballos.
- Él puede vender ahora los dos caballos. La ganacia total será $2 \cdot Y[1] = 8$.

Luego, suponga que hay $M = 1$ actualización: cambiar $Y[1]$ a 2 .

Después de la actualización tendremos:

| | | | |
|---|---|---|---|
| | 0 | 1 | 2 |
| X | 2 | 1 | 3 |
| Y | 3 | 2 | 1 |

En este caso, una de las soluciones óptimas es vender un caballo después del año 0 y luego tres caballos en el año 2.

Todo el proceso se ve como sigue:

- Inicialmente Mansur tiene 1 caballo.
- Después del año 0, él tendrá $1 \cdot X[0] = 2$ caballos.
- El puede ahora vender uno de esos caballos, por $Y[0] = 3$, y le queda un caballo.
- Después de 1 año, él tendrá $1 \cdot X[1] = 1$ caballo.
- Después de 2 años, él tendrá $1 \cdot X[2] = 3$ caballos.
- Él puede vender ahora tres caballos por $3 \cdot Y[2] = 3$. La cantidad total de dinero es $3 + 3 = 6$.

Tarea

A usted le dan N , X , Y y una lista de actualizaciones. Antes de la primera actualización y después de cada actualización, calcule la cantidad máxima que Mansur podría haber obtenido módulo $10^9 + 7$.

Usted necesita implementar las funciones `init`, `updateX` y `updateY`.

- `init(N, X, Y)` — El calificador llamará esta función primero exactamente una vez.
 - N : el número de años.
 - X : un arreglo de longitud N . Para $0 \leq i \leq N - 1$, $X[i]$ da el coeficiente de crecimiento para el año i .
 - Y : un arreglo de longitud N . Para $0 \leq i \leq N - 1$, $Y[i]$ da el precio de un caballo después del año i .
 - Note que ambos X y Y especifican los valores iniciales dados por Mansur (antes de cualquier actualización).
 - Después de que `init` retorne, los arreglos X y Y siguen siendo válidos, y usted puede modificar su contenido si lo desea.
 - La función debe devolver la cantidad máxima de dinero que Mansur puede hacer después

de esta actualización, módulo $10^9 + 7$.

- `updateX(pos, val)`
 - `pos`: un entero en el rango $0, \dots, N - 1$.
 - `val`: el nuevo valor para $X[pos]$.
 - La función debe devolver la cantidad máxima de dinero que Mansur puede hacer después de esta actualización, módulo $10^9 + 7$.
- `updateY(pos, val)`
 - `pos`: un entero en el rango $0, \dots, N - 1$.
 - `val`: el nuevo valor para $Y[pos]$.
 - La función debe devolver la cantidad máxima de dinero que Mansur puede hacer después de esta actualización, módulo $10^9 + 7$.

Usted puede asumir que todos los valores iniciales, así como los valores actualizados de $X[i]$ y $Y[i]$ están entre 1 y 10^9 inclusive.

Después de llamar a `init`, el calificador llamará `updateX` y `updateY` varias veces. El número total de llamadas a `updateX` y `updateY` será M .

Subtareas

| subtarea | puntos | N | M | restricciones adicionales |
|----------|--------|-------------------------|-------------------------|--|
| 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 | $1 \leq N \leq 1,000$ | $0 \leq M \leq 1,000$ | nada |
| 3 | 20 | $1 \leq N \leq 500,000$ | $0 \leq M \leq 100,000$ | $X[i] \geq 2$ y $val \geq 2$ para <code>init</code> y <code>updateX</code> respectivamente |
| 4 | 23 | $1 \leq N \leq 500,000$ | $0 \leq M \leq 10,000$ | nada |
| 5 | 23 | $1 \leq N \leq 500,000$ | $0 \leq M \leq 100,000$ | nada |

Calificador de ejemplo

El calificador de ejemplo lee la entrada del archivo `horses.in` en el siguiente formato:

- línea 1: N
- línea 2: $X[0] \dots X[N - 1]$
- línea 3: $Y[0] \dots Y[N - 1]$
- línea 4: M
- líneas 5, ..., $M + 4$: tres números `type pos val` (`type=1` para `updateX` y `type=2` para `updateY`).

El calificador ejemplo imprime el valor de retorno de `init` seguido por los valores de retorno para todas las llamadas de `updateX` y `updateY`.