

Actualiza y vencerás*Contribución de David Lescano***Descripción del problema**

Gennady se encuentra en la casa de su mejor amigo. Ambos se pasan todo el día jugando juegos de mesa, ya que una vez que empiezan son muy competitivos y no paran hasta obtener el resultado que quieren.

Valentina, la mamá de su amigo, se da cuenta de esto, y no le gusta para nada ¡No se enfocan en sus estudios! Decide entonces inventar un juego que no conocen, con la idea de que se aburran rápidamente y se enfoquen en estudiar.

El juego se inicia con N números en hilera, cuyas posiciones se numeran desde 1 hasta N . A continuación ocurren sucesivamente Q eventos. Hay dos tipos de eventos:

1. Valentina **reemplaza** un número x de la hilera, por un nuevo número y , que pasa a ocupar la misma posición en la que antes estaba x .
2. Valentina pide a uno de los participantes calcular dos cosas:
 - La **diferencia entre el máximo y el mínimo** de los números **desde** la posición **L** de la hilera **hasta** la posición **R**, inclusive.
 - La cantidad de números **distintos menores** a K , en **toda** la hilera (es decir considerando desde la posición 1 hasta la N inclusive).

Debes ayudar a Valentina a obtener las respuestas correctas, de forma que pueda saber rápidamente si ellos contestan bien o mal.

Detalles de implementación

Se debe implementar la función `actualiza(hilera, t,L,R,K, diferencia, distintosMenorK)`, que recibe:

- `hilera`: un arreglo de N enteros, con los números de la hilera en orden.
- `t,L,R,K`: arreglos de Q enteros, que especifican los eventos que ocurren. Para el evento i -ésimo ($0 \leq i < Q$):
 1. Si es de tipo 1, será $t[i] = 1$, $K[i] = 0$ y el evento consiste en que Valentina reemplaza al número en la posición $L[i]$ de la hilera, por el nuevo número $R[i]$.
 2. Si es de tipo 2, será $t[i] = 2$, y el evento consiste en que Valentina pide calcular las dos cosas explicadas anteriormente. $L[i]$, $R[i]$ y $K[i]$ indican los correspondientes valores L , R , K para el cálculo como se explicó antes.

Se debe guardar la respuesta en los arreglos `diferencia` y `distintosMenorK`, colocando en ellos tantos elementos como eventos de tipo 2 (cálculos de valores) haya, con las respuestas en orden.

Evaluador local

El evaluador lee de la entrada estándar:

- Una línea con dos enteros N y Q
- Una línea con N enteros `hilera[i]`
- Q líneas, cada una con 4 enteros $t[i]$, $L[i]$, $R[i]$, $K[i]$

Devuelve por la salida estándar una primera línea con el contenido del arreglo `diferencia`, y una segunda línea con el contenido del arreglo `distintosMenorK`.

Restricciones

- $1 \leq N, Q \leq 200.000$
- $1 \leq hilera[i] \leq 1.000.000.000$
- $1 \leq t[i] \leq 2$
- Si $t[i] = 1$, es $K[i] = 0$
- Si $t[i] = 2$, es $1 \leq K[i] \leq 1.000.000.000$
- $1 \leq L[i] \leq N$
- Si $t[i] = 1$, es $1 \leq R[i] \leq N$
- Si $t[i] = 2$, es $1 \leq R[i] \leq 1.000.000.000$

Ejemplos

Si el evaluador local recibe la siguiente entrada:

```
10 5
2 8 7 3 9 2 10 3 3 7
2 4 5 11
2 4 5 10
1 7 9 0
2 1 10 11
2 2 4 100
```

Una implementación correcta debe devolver:

```
6 6 7 5
6 5 5 5
```

Puntuación

- Se recibe el 50% de los puntos por cada uno de los arreglos diferencia y distintosMenorK que esté correctamente calculado en toda la subtarea.

Subtareas

1. $1 \leq N, Q \leq 1.000$ (6 puntos)
2. $t[i] = 2$ para todo i (24 puntos)
3. $1 \leq hilera[i], R[i] \leq N$ (30 puntos)
4. Sin más restricciones (40 puntos)