

## Comparando plantas (plants)

Hazel el botánico visitó una exhibición especial en el Jardín Botánico de Singapur. En esta exhibición,  $n$  plantas **todas de diferentes alturas** se colocan alrededor de un círculo. Estas plantas se numeran desde 0 hasta  $n - 1$  en el sentido de las agujas del reloj, con la planta  $n - 1$  al lado de la planta 0.

Para cada planta  $i$  ( $0 \leq i \leq n - 1$ ), Hazel comparó la planta  $i$  con cada una de las siguientes  $k - 1$  plantas en el sentido de las agujas del reloj, y anotó el número  $r[i]$  que denota cuántas de estas  $k - 1$  plantas son más altas que la planta  $i$ . Así, cada valor  $r[i]$  depende de las alturas relativas de un conjunto de  $k$  plantas consecutivas.

Por ejemplo, supongamos que  $n = 5$ ,  $k = 3$  y  $i = 3$ . Las siguientes  $k - 1 = 2$  plantas en el sentido de las agujas del reloj desde la planta  $i = 3$  serían la planta 4 y la planta 0. Si la planta 4 fuera más alta que la planta 3 y la planta 0 fuera más baja que la planta 3, Hazel anotaría  $r[3] = 1$ .

Puedes asumir que Hazel registró los valores  $r[i]$  correctamente. Por lo tanto, existe al menos una configuración de alturas diferentes para las plantas que es consistente con estos valores.

Se te ha solicitado que compares las alturas de  $q$  pares de plantas. Lamentablemente, no tienes acceso a la exhibición. Tu única fuente de información es el cuaderno de Hazel, que contiene el valor  $k$  y la secuencia de valores  $r[0], \dots, r[n - 1]$ .

Por cada par de plantas diferentes  $x$  e  $y$  que se deben comparar, debes determinar cuál de las siguientes tres situaciones ocurre:

- La planta  $x$  es definitivamente más alta que la planta  $y$ : en cualquier configuración de alturas diferentes  $h[0], \dots, h[n - 1]$  consistente con el arreglo  $r$  se tiene que  $h[x] > h[y]$ .
- La planta  $x$  es definitivamente más baja que la planta  $y$ : en cualquier configuración de alturas diferentes  $h[0], \dots, h[n - 1]$  consistente con el arreglo  $r$  se tiene que  $h[x] < h[y]$ .
- La comparación no es concluyente: no se da ninguno de los dos casos anteriores.

## Detalles de implementación

Debes implementar las siguientes funciones:

```
void init(int k, int[] r)
```

- $k$ : la cantidad de plantas consecutivas cuyas alturas determinan cada valor individual  $r[i]$ .
- $r$ : un arreglo de tamaño  $n$ , en el que  $r[i]$  es la cantidad de plantas más altas que la planta  $i$  entre las siguientes  $k - 1$  plantas en el sentido de las agujas del reloj.

- Esta función es llamada exactamente una vez, antes de cualquier llamada a `compare_plants`.

```
int compare_plants(int x, int y)
```

- $x, y$ : números de las plantas que se deben comparar.
- Esta función debe retornar:
  - 1 si la planta  $x$  es definitivamente más alta que la planta  $y$ ,
  - $-1$  si la planta  $x$  es definitivamente más baja que la planta  $y$ ,
  - 0 si la comparación no es concluyente.
- Esta función es llamada exactamente  $q$  veces.

## Ejemplos

### Ejemplo 1

Considere la siguiente llamada:

```
init(3, [0, 1, 1, 2])
```

Supongamos que el evaluador realiza la llamada `compare_plants(0, 2)`. Como  $r[0] = 0$  podemos deducir inmediatamente que la planta 2 no es más alta que la planta 0. Por lo tanto, la llamada debe retornar 1.

Supongamos que a continuación el evaluador realiza la llamada `compare_plants(1, 2)`. Para todas las posibles configuraciones de alturas que satisfacen las restricciones anteriormente explicadas, la planta 1 es más baja que la planta 2. Por lo tanto, la llamada debe retornar  $-1$ .

### Ejemplo 2

Considere la siguiente llamada:

```
init(2, [0, 1, 0, 1])
```

Supongamos que el evaluador realiza la llamada `compare_plants(0, 3)`. Como  $r[3] = 1$ , sabemos que la planta 0 es más alta que la planta 3. Por lo tanto, la llamada debe retornar 1.

Supongamos que a continuación el evaluador realiza la llamada `compare_plants(1, 3)`. Las dos configuraciones de alturas  $[3, 1, 4, 2]$  y  $[3, 2, 4, 1]$  son ambas consistentes con las mediciones de Hazel. Como la planta 1 es más baja que la planta 3 en una configuración y más alta que la planta 3 en la otra, esta llamada debe retornar 0.

## Cotas

- $2 \leq k \leq n \leq 200\,000$
- $1 \leq q \leq 200\,000$
- $0 \leq r[i] \leq k - 1$  (para todo  $0 \leq i \leq n - 1$ )
- $0 \leq x < y \leq n - 1$
- Existe una o más configuraciones **de alturas diferentes** de plantas consistentes con el arreglo  $r$ .

## Subtareas

1. (5 puntos)  $k = 2$
2. (14 puntos)  $n \leq 5000$ ,  $2 \cdot k > n$
3. (13 puntos)  $2 \cdot k > n$
4. (17 puntos) La respuesta correcta para cada llamada a `compare_plants` es 1 o  $-1$ .
5. (11 puntos)  $n \leq 300$ ,  $q \leq \frac{n \cdot (n-1)}{2}$
6. (15 puntos)  $x = 0$  en cada llamada a `compare_plants`.
7. (25 puntos) Sin más restricción.

## Evaluador local

El evaluador local lee la entrada con el siguiente formato:

- línea 1:  $n \ k \ q$
- línea 2:  $r[0] \ r[1] \ \dots \ r[n - 1]$
- línea  $3 + i$  ( $0 \leq i \leq q - 1$ ):  $x \ y$  para la  $i$ -ésima llamada a `compare_plants`

El evaluador local imprime tus respuestas con el siguiente formato:

- línea  $1 + i$  ( $0 \leq i \leq q - 1$ ): el valor retornado por la  $i$ -ésima llamada a `compare_plants`.