



## Comparación de Plantas (plants)

Hazel, la botánica, visitó una exposición especial en los Jardines Botánicos de Singapur. En esta exposición,  $n$  plantas de **alturas distintas** son sembradas en un círculo. Estas plantas son etiquetadas desde 0 hasta  $n - 1$  en el orden de las agujas del reloj, con la planta  $n - 1$  al lado de la planta 0.

Por 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 orden de las agujas del reloj, y anotó el número  $r[i]$  denotando cuántas de estas  $k - 1$  plantas son más altas que la planta  $i$ . Por lo tanto, cada valor  $r[i]$  depende de las alturas relativas de  $k$  plantas consecutivas.

Por ejemplo, supóngase 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 pequeña que la planta 3, Hazel debería escribir  $r[3] = 1$ .

Puedes suponer que Hazel anotó los valores  $r[i]$  correctamente. Por lo tanto, hay al menos una configuración de alturas distintas de las plantas consistente con estos valores.

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

Por cada par de plantas diferentes  $x$  y  $y$  que necesiten ser comparadas, determina cuál de las siguientes situaciones ocurre:

- La planta  $x$  es definitivamente más alta que la planta  $y$ : en cualquier configuración de alturas distintas  $h[0], \dots, h[n - 1]$  consistente con el arreglo  $r$  tenemos que  $h[x] > h[y]$ .
- La Planta  $x$  es definitivamente más pequeña que la planta  $y$ : en cualquier configuración de alturas distintas  $h[0], \dots, h[n - 1]$  consistente con el arreglo  $r$  tenemos que  $h[x] < h[y]$ .
- La comparación es inconclusa: ninguno de los dos casos previos aplica.

## Detalles de implementación

Debes implementar los siguientes procedimientos:

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

- $k$ : el número de plantas consecutivas cuyas alturas determinan cada valor individual  $r[i]$ .
- $r$ : un arreglo de tamaño  $n$ , donde  $r[i]$  es el número de plantas más altas que la planta  $i$  entre

las siguientes  $k - 1$  plantas en el sentido de las agujas del reloj.

- Este procedimiento se llama exactamente una vez, antes de cualquier llamada a `compare_plants`.

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

- $x, y$ : etiquetas de las plantas a ser comparadas.
- Este procedimiento debe retornar:
  - 1 si la planta  $x$  es definitivamente más alta que la planta  $y$ ,
  - $-1$  si la planta  $x$  es definitivamente más pequeña que la planta  $y$ ,
  - 0 si la comparación es inconclusa.
- Este procedimiento es llamado exactamente  $q$  veces.

## Ejemplos

### Ejemplo 1

Considera la siguiente llamada:

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

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

Supongamos que el evaluador luego llama a `compare_plants(1, 2)`. Para todas las configuraciones posibles de alturas que se ajustan a las restricciones anteriores, la planta 1 es más pequeña que la planta 2. Por lo tanto, la llamada debería retornar  $-1$ .

### Ejemplo 2

Considera la siguiente llamada:

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

Supongamos que el evaluador llama a `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 debería retornar 1.

Supongamos que el evaluador luego llama `compare_plants(1, 3)`. Dos configuraciones de alturas  $[3, 1, 4, 2]$  y  $[3, 2, 4, 1]$  son ambas consistentes con las medidas de Hazel. Como la planta 1 es más pequeña que la planta 3 en una configuración y más alta que la 3 en la otra, esta llamada debería retornar 0.

## Restricciones

- $2 \leq k \leq n \leq 200\,000$
- $1 \leq q \leq 200\,000$
- $0 \leq r[i] \leq k - 1$  (para todo  $0 \leq i \leq n - 1$ )
- $0 \leq x < y \leq n - 1$
- Existe una o más configuraciones de **alturas distintas** 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 a cada llamada de `compare_plants` es 1 o  $-1$ .
5. (11 puntos)  $n \leq 300$ ,  $q \leq \frac{n \cdot (n-1)}{2}$
6. (15 puntos)  $x = 0$  por cada llamada de `compare_plants`.
7. (25 puntos) Sin restricciones adicionales.

## Evaluador de ejemplo

El evaluador de ejemplo lee la entrada en 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 de ejemplo imprime tu respuesta en el siguiente formato:

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