

La pianta più alta (plants)

Hazel sta visitando una speciale esibizione dei *Singapore Botanical Gardens*. In questa esibizione n piante di **altezze distinte** sono disposte in cerchio, numerate da 0 a $n - 1$ in ordine orario, con la pianta $n - 1$ accanto alla pianta 0.

Per ogni $i = 0 \dots n - 1$, Hazel confronta la pianta i con le successive $k - 1$ in ordine orario, calcolando il numero $r[i]$ di queste che sono più alte della pianta i . Per esempio, supponiamo che $n = 5$, $k = 3$, $i = 3$; per cui le successive $k - 1 = 2$ piante in ordine orario dalla pianta $i = 3$ sono le piante 4 e 0. Se la pianta 4 è più alta della 3 mentre la pianta 0 è più bassa della 3, Hazel calcolerà $r[3] = 1$.

Devi confrontare le altezze di q coppie di piante, senza accedere all'esibizione e conoscendo solamente il valore k e la sequenza $r[0], \dots, r[n - 1]$. Per ogni coppia di piante distinte x e y da confrontare, determina se:

- La pianta x è sicuramente più alta della pianta y : $h[x] > h[y]$ in ogni configurazione di altezze distinte $h[0], \dots, h[n - 1]$ consistente con l'array r .
- La pianta x è sicuramente più bassa della pianta y : $h[x] < h[y]$ in ogni configurazione di altezze distinte $h[0], \dots, h[n - 1]$ consistente con l'array r .
- Il confronto è inconcludente: ci sono configurazioni consistenti sia con $h[x] > h[y]$ che con $h[x] < h[y]$.

Dato che Hazel è molto attenta, puoi assumere che esista almeno una configurazione valida di altezze distinte consistente con i valori $r[i]$.

Note di implementazione

Devi implementare le seguenti funzioni:

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

- k : il numero di piante consecutive considerato nel calcolo di un $r[i]$.
- r : un array di dimensione n , dove $r[i]$ è il numero di piante più alte della pianta i tra le successive $k - 1$ in senso orario.
- La funzione viene chiamata esattamente una volta, prima di ogni chiamata a `compare_plants`.

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

- x, y : i numeri delle piante da confrontare.
- Questa funzione deve restituire:
 - 1 se la pianta x è sicuramente più alta della pianta y ,
 - -1 se la pianta x è sicuramente più bassa della pianta y ,
 - 0 se il confronto è inconcludente.
- Questa funzione viene chiamata esattamente q volte.

Esempi

Esempio 1

Considera la seguente chiamata:

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

Supponi che il grader chiami `compare_plants(0, 2)`. Dato che $r[0] = 0$, la pianta 2 non può essere più alta della pianta 0, per cui la funzione deve restituire 1.

Supponi che il grader chiami `compare_plants(1, 2)`. In ogni configurazione di altezze che rispetta le condizioni, la pianta 1 è più bassa della pianta 2, per cui la funzione deve restituire -1 .

Esempio 2

Considera la seguente chiamata:

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

Supponi che il grader chiami `compare_plants(0, 3)`. Dato che $r[3] = 1$ sappiamo che la pianta 0 è più alta della pianta 3, per cui la funzione deve restituire 1.

Supponi che il grader chiami `compare_plants(1, 3)`. Le configurazioni di altezze $[3, 1, 4, 2]$ e $[3, 2, 4, 1]$ sono entrambe consistenti con le misurazioni Hazel. Dato che la pianta 1 è più bassa della 3 in una configurazione e più alta nell'altra, la funzione deve restituire 0.

Assunzioni

- $2 \leq k \leq n \leq 200\,000$
- $1 \leq q \leq 200\,000$
- $0 \leq r[i] \leq k - 1$ (per ogni $0 \leq i \leq n - 1$)
- $0 \leq x < y \leq n - 1$
- Esiste almeno una configurazione di **altezze distinte** consistente con l'array r .

Subtask

1. (5 punti) $k = 2$
2. (14 punti) $n \leq 5000, 2 \cdot k > n$
3. (13 punti) $2 \cdot k > n$
4. (17 punti) La risposta corretta di ogni chiamata a `compare_plants` è 1 o -1.
5. (11 punti) $n \leq 300, q \leq \frac{n \cdot (n-1)}{2}$
6. (15 punti) $x = 0$ per ogni chiamata a `compare_plants`.
7. (25 punti) Nessuna limitazione aggiuntiva.

Grader di esempio

Il grader di esempio legge l'input nel seguente formato:

- riga 1: $n \ k \ q$
- riga 2: $r[0] \ r[1] \ \dots \ r[n-1]$
- riga $3 + i$ ($0 \leq i \leq q-1$): $x \ y$ per l' i -esima chiamata a `compare_plants`

Il grader di esempio stampa l'output nel seguente formato:

- riga $1 + i$ ($0 \leq i \leq q-1$): il valore restituito dalla i -esima chiamata a `compare_plants`.