

## Membandingkan Tanaman (plants)

Hazel sedang mengunjungi pameran spesial di Kebun Raya Singapore. Dalam pameran ini,  $n$  tanaman dengan **ketinggian berbeda-beda** diletakkan dalam bentuk melingkar. Tanaman ini diberi nomor dari 0 hingga  $n - 1$  searah jarum jam, dengan tanaman  $n - 1$  di sebelah tanaman 0.

Untuk setiap tanaman  $i$  ( $0 \leq i \leq n - 1$ ), Hazel membandingkan tanaman  $i$  dengan setiap tanaman  $k - 1$  berikutnya searah jarum jam, dan menulis nomor  $r[i]$  yang menunjukkan banyaknya tanaman dari  $k - 1$  tanaman tersebut yang lebih tinggi dari tanaman  $i$ . Jadi, setiap nilai dari  $r[i]$  bergantung pada ketinggian relatif dari  $k$  tanaman berturut-turut.

Misalkan  $n = 5$ ,  $k = 3$ , dan  $i = 3$ . Sebanyak  $k - 1 = 2$  tanaman berikutnya yang searah jarum jam dari tanaman  $i = 3$  adalah tanaman 4 dan tanaman 0. Jika tanaman 4 lebih tinggi daripada tanaman 3 dan tanaman 0 lebih rendah daripada tanaman 3, maka Hazel akan menulis  $r[3] = 1$ .

Hazel selalu menulis nilai dari  $r[i]$  dengan benar. Jadi, setidaknya ada satu konfigurasi dengan setiap tanaman memiliki ketinggian yang berbeda-beda yang memenuhi nilai-nilai tersebut.

Anda diminta untuk membandingkan tinggi dari  $q$  pasang tanaman. Namun, Anda tidak memiliki akses ke pameran tersebut. Satu-satunya informasi yang Anda punya adalah catatan Hazel yang berisikan nilai dari  $k$  dan barisan yang berisikan nilai dari  $r[0], \dots, r[n - 1]$ .

Untuk setiap pasangan dari tanaman berbeda  $x$  dan  $y$  yang perlu dibandingkan, tentukan manakah di antara ketiga situasi berikut yang terjadi:

- Tanaman  $x$  pasti lebih tinggi dari tanaman  $y$ : dalam konfigurasi apapun  $h[0], \dots, h[n - 1]$  akan selalu berlaku  $h[x] > h[y]$ .
- Tanaman  $x$  pasti lebih rendah dari tanaman  $y$ : dalam konfigurasi apapun  $h[0], \dots, h[n - 1]$  akan selalu berlaku  $h[x] < h[y]$ .
- Tidak dapat ditentukan: 2 kondisi di atas tidak terpenuhi.

## Implementasi

Anda diminta untuk membuat fungsi-fungsi:

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

- $k$ : banyaknya tanaman berurutan yang ketinggian-ketinggiannya menentukan nilai  $r[i]$ .
- $r$ : barisan bilangan dengan panjang  $n$  dengan  $r[i]$  adalah banyaknya tanaman yang lebih tinggi daripada tanaman  $i$  diantara  $k - 1$  tanaman berikutnya.

- fungsi ini akan dipanggil tepat sekali, sebelum pemanggilan `compare_plants`.

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

- $x, y$ : nomor dari dua tanaman yang ingin dibandingkan.
- Fungsi ini harus mengembalikan nilai:
  - 1 jika tanaman  $x$  pasti lebih tinggi daripada tanaman  $y$ ,
  - $-1$  jika tanaman  $x$  pasti lebih rendah daripada tanaman  $y$ ,
  - 0 jika tidak dapat ditentukan.
- Fungsi ini akan dipanggil tepat  $q$  kali.

## Contoh

### Contoh 1

Misalkan pemanggilan fungsi:

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

Misalkan *grader* melakukan pemanggilan `compare_plants(0, 2)`. Karena  $r[0] = 0$  kita dapat memastikan bahwa tanaman 2 pasti tidak lebih tinggi daripada tanaman 0. Maka, fungsi harus mengembalikan nilai 1.

Misalkan *grader* melakukan pemanggilan `compare_plants(1, 2)`. Untuk setiap konfigurasi yang memenuhi, tanaman 1 pasti lebih rendah daripada tanaman 2. Maka, fungsi harus mengembalikan nilai  $-1$ .

### Contoh 2

Misalkan pemanggilan fungsi:

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

Misalkan *grader* melakukan pemanggilan `compare_plants(0, 3)`. Karena  $r[3] = 1$ , kita tahu tanaman 0 lebih tinggi daripada tanaman 3. Maka, fungsi harus mengembalikan nilai 1.

Misalkan *grader* melakukan pemanggilan `compare_plants(1, 3)`. Dua konfigurasi dengan tinggi  $[3, 1, 4, 2]$  dan  $[3, 2, 4, 1]$  dapat memenuhi catatan Hazel. Karena tanaman 1 lebih rendah daripada tanaman 3 di satu konfigurasi dan lebih tinggi daripada tanaman 3 di konfigurasi lainnya, maka fungsi harus mengembalikan nilai 0.

## Batasan

- $2 \leq k \leq n \leq 200\,000$

- $1 \leq q \leq 200\,000$
- $0 \leq r[i] \leq k - 1$  (untuk setiap  $0 \leq i \leq n - 1$ )
- $0 \leq x < y \leq n - 1$
- Dijamin pasti ada satu atau lebih konfigurasi dengan **ketinggian yang berbeda-beda** dan memenuhi syarat barisan  $r$ .

## Subsoal

1. (5 poin)  $k = 2$
2. (14 poin)  $n \leq 5000, 2 \cdot k > n$
3. (13 poin)  $2 \cdot k > n$
4. (17 poin) Jawaban dari setiap pemanggilan `compare_plants` pasti 1 atau  $-1$ .
5. (11 poin)  $n \leq 300, q \leq \frac{n \cdot (n-1)}{2}$
6. (15 poin)  $x = 0$  untuk setiap pemanggilan `compare_plants`.
7. (25 poin) Tidak ada batasan tambahan.

## Contoh *grader*

Contoh *grader* membaca masukkan dengan format sebagai berikut:

- baris 1:  $n \ k \ q$
- baris 2:  $r[0] \ r[1] \ \dots \ r[n - 1]$
- baris  $3 + i$  ( $0 \leq i \leq q - 1$ ):  $x \ y$  untuk pemanggilan `compare_plants` ke- $i$ .

Contoh *grader* mencetak jawaban dengan format sebagai berikut:

- baris  $1 + i$  ( $0 \leq i \leq q - 1$ ): nilai dari pemanggilan `compare_plants` ke- $i$ .