

## Porównywanie roślin (plants)

Botaniczka Hazel odwiedziła wystawę w singapurskich Ogrodach Botanicznych. Znajduje się tam  $n$  roślin **różnej wysokości** rozmieszczonych na okręgu. Rośliny te są numerowane od 0 do  $n - 1$ , zgodnie z kierunkiem wskazówek zegara, przy czym po roślinie  $n - 1$  mamy roślinę 0.

Dla każdej z roślin  $i$  ( $0 \leq i \leq n - 1$ ), Hazel porównuje roślinę  $i$  z każdą z  $k - 1$  roślin następujących po niej zgodnie z kierunkiem wskazówek zegara, a następnie zapisuje jako  $r[i]$  liczbę roślin z tej grupy  $k - 1$  roślin, które są wyższe od rośliny  $i$ . Zatem każda z wartości  $r[i]$  zależy od względnej wysokości pewnych kolejnych  $k$  roślin.

Na przykład, dla  $n = 5$ ,  $k = 3$  oraz  $i = 3$ , kolejne  $k - 1 = 2$  rośliny od rośliny o numerze  $i = 3$  miałyby numery 4 i 0. Gdyby roślina 4 była wyższa od rośliny 3, a roślina 0 niższa od rośliny 3, to Hazel zanotowałaby  $r[3] = 1$ .

Możesz założyć, że Hazel nie zrobiła błędu i wszystkie wartości  $r[i]$  są poprawne. Zatem co najmniej jedna konfiguracja różnych wysokości roślin generuje takie wyniki, jak w notesie Hazel.

Twoim zadaniem jest porównanie wysokości  $q$  par roślin. Niestety nie masz dostępu do wystawy. Twoim jedynym źródłem informacji jest notes Hazel, w którym znajduje się wartość  $k$  i wartości  $r[0], \dots, r[n - 1]$ .

Dla każdej z różnych par  $x$  i  $y$ , które porównujesz, musisz określić, która z następujących sytuacji ma miejsce:

- Roślina  $x$  jest na pewno wyższa od rośliny  $y$ : w każdej konfiguracji różnych wysokości  $h[0], \dots, h[n - 1]$  zgodnych z wartościami z tablicy  $r$ , mamy  $h[x] > h[y]$ .
- Roślina  $x$  jest na pewno niższa od rośliny  $y$ : w każdej konfiguracji różnych wysokości  $h[0], \dots, h[n - 1]$  zgodnych z wartościami z tablicy  $r$ , mamy  $h[x] < h[y]$ .
- Nie sposób jednoznacznie określić, który z wyżej opisanych przypadków zachodzi.

## Szczegóły implementacji

Powinieneś napisać następujące procedury:

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

- $k$ : liczba kolejnych roślin, których wysokości są używane do wyznaczenia wszystkich wartości  $r[i]$ .
- $r$ : tablica długości  $n$ , gdzie  $r[i]$  jest liczbą roślin pośród kolejnych  $k - 1$  roślin, które są wyższe niż roślina  $i$ .

- Ta procedura jest wywoływana dokładnie raz, zanim użyje się jakiegokolwiek wywołania funkcji `compare_plants`.

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

- $x, y$ : numery roślin, które mają być porównywane.
- Procedura ta powinna zwrócić:
  - 1, jeśli roślina  $x$  jest na pewno wyższa niż roślina  $y$ ,
  - $-1$ , jeśli roślina  $x$  jest na pewno niższa niż roślina  $y$ ,
  - 0, jeśli nie da się tego jednoznacznie ustalić.
- Procedura ta będzie wywołana dokładnie  $q$  razy.

## Przykłady

### Przykład 1

Rozważmy następujące wywołanie:

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

Założmy, że program sprawdzający wywołuje `compare_plants(0, 2)`. Ponieważ  $r[0] = 0$ , możemy od razu stwierdzić, że roślina 2 nie jest wyższa niż roślina 0. W związku z tym wynikiem wywołania powinno być 1.

Założmy, że program sprawdzający następnie wywołuje `compare_plants(1, 2)`. Dla wszystkich możliwych konfiguracji wysokości zgodnych z powyższymi danymi roślina 1 jest niższa niż roślina 2. Wynikiem powinno być zatem  $-1$ .

### Przykład 2

Rozważmy następujące wywołanie:

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

Gdy program sprawdzający wywoła `compare_plants(0, 3)`, ze względu na to, że  $r[3] = 1$  wiemy, że roślina 0 jest wyższa niż roślina 3. W związku z tym wynikiem jest 1.

Założmy teraz, że program sprawdzający wywołuje `compare_plants(1, 3)`. Są co najmniej dwie konfiguracje wysokości:  $[3, 1, 4, 2]$  i  $[3, 2, 4, 1]$  zgodne z pomiarami Hazel. Ponieważ roślina 1 jest niższa niż 3 w jednej z nich i wyższa w drugiej, wynikiem powinno być 0.

## Ograniczenia

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

- $1 \leq q \leq 200\,000$
- $0 \leq r[i] \leq k - 1$  (dla  $0 \leq i \leq n - 1$ )
- $0 \leq x < y \leq n - 1$
- Powinna istnieć co najmniej jedna konfiguracja **różnych wysokości** roślin zgodna z tablicą  $r$ .

## Podzadania

1. (5 punktów)  $k = 2$
2. (14 punktów)  $n \leq 5000, 2 \cdot k > n$
3. (13 punktów)  $2 \cdot k > n$
4. (17 punktów) Prawidłową odpowiedzią na wszystkie wywołania `compare_plants` jest 1 lub  $-1$ .
5. (11 punktów)  $n \leq 300, q \leq \frac{n \cdot (n-1)}{2}$
6. (15 punktów)  $x = 0$  dla każdego z wywołań `compare_plants`.
7. (25 punktów) Brak dodatkowych ograniczeń.

## Przykładowy program sprawdzający

Przykładowy program sprawdzający wczytuje wejście w następującym formacie:

- wiersz 1:  $n \ k \ q$
- wiersz 2:  $r[0] \ r[1] \ \dots \ r[n-1]$
- wiersze  $3 + i$  ( $0 \leq i \leq q - 1$ ):  $x \ y$  dla  $i$ -tego wywołania `compare_plants`

Przykładowa sprawdzaczka wypisuje wyjście w następującym formacie:

- wiersze  $1 + i$  ( $0 \leq i \leq q - 1$ ): zawiera wynik  $i$ -tego wywołania `compare_plants`.