

Növények

Hazel a szingapúri botanikus kertbe látogat, ahol n **különböző magasságú** növényt helyeztek el kör alakban, az óramutató járása szerinti sorrendben, 0-tól sorszámozva.

Hazel minden növényt összehasonlított az óramutató járása szerint mögötte levő $k - 1$ növénnel és leírta ($r[i]$), hogy közülük hánynál magasabb.

Például, ha $n = 5$, $k = 3$ és $i = 3$, akkor az óramutató járása szerint következő $k - 1 = 2$ növény a 4. és a 0. Ha a 4. magasabb, mint a 3., a 0. pedig alacsonyabb, akkor $r[3] = 1$.

Biztosan van legalább egy olyan konfigurációja a magasságoknak, amely megfelel az $r[i]$ értékeknek, de a magasságokat nem ismerjük.

Programoddal q növényt kell összehasonlítanod, de ehhez csak a k és az $r[0], \dots, r[n - 1]$ értékek állnak rendelkezésedre.

Minden x és y növénypárra határozd meg, hogy az alábbi három eset közül melyik teljesül:

- Az x . magasabb y -nál: ha bármely az r értékeinek megfelelő $h[0], \dots, h[n - 1]$ magasságok esetén $h[x] > h[y]$.
- Az x alacsonyabb y -nál: ha bármely az r értékeinek megfelelő $h[0], \dots, h[n - 1]$ magasságok esetén $h[x] < h[y]$.
- Nem állapítható meg egyértelműen, hogy melyik magasabb.
- Nem állapítható meg egyértelműen, hogy melyik magasabb, azaz van olyan r -nak megfelelő magasságsorozat, amely szerint az x . a magasabb és van olyan is, amely szerint pedig az x . az alacsonyabb.

Megvalósítás

A következő függvényeket kell elkészítened:

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

- k : az egymásutáni növények száma, ami alapján az $r[i]$ értékek keletkeztek.
- r : n elemű tömb, ahol $r[i]$ az i . növénynél magasabbak száma az óramutató járása szerint következő $k - 1$ közül.
- Az értékelő ezt függvényt csak egyszer hívja, a `compare_plants` hívások előtt.

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

- x, y : a két összehasonlítandó növény sorszáma.
- A függvény értéke:
 - 1, ha az x . növény magasabb az y -nél,
 - -1 , ha az x . növény alacsonyabb az y -nél,
 - 0, ha nem állapítható meg egyértelműen, hogy melyik magasabb.
- Ezt a függvényt pontosan q -szor hívják.

Példák

1. példa

Az `init` hívása:

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

Ha az értékelő a `compare_plants(0, 2)`-t hívja, akkor mivel $r[0] = 0$, ezért tudjuk, hogy a 2. növény nem magasabb, mint a 0. növény, tehát a függvény az 1 értéket adja eredményül.

Ha a következő hívás a `compare_plants(1, 2)`, akkor a feltételnek megfelelő minden lehetséges magasságok esetén az 1. alacsonyabb a 2.-nél, tehát a függvény a -1 értéket adja eredményül.

2. példa

Az `init` hívása:

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

Ha az értékelő a `compare_plants(0, 3)`-t hívja, mivel $r[3] = 1$, ezért tudjuk, hogy a 0. magasabb a 3.-nál, tehát a függvény az 1 értéket adja eredményül.

Ha a következő hívás a `compare_plants(1, 3)`, akkor a következő két magasságsorozat mindegyike ($[3, 1, 4, 2]$ és $[3, 2, 4, 1]$) megfelel a Hazel által mért értékeknek. Az első esetben az 1. alacsonyabb a 3.-nál, a második szerint pedig magasabb, tehát a függvény a 0 értéket adja eredményül.

Korlátok

- $2 \leq k \leq n \leq 200\,000$
- $1 \leq q \leq 200\,000$
- $0 \leq r[i] \leq k - 1$ ($0 \leq i \leq n - 1$)
- $0 \leq x < y \leq n - 1$
- Biztosan van legalább egy olyan konfigurációja a magasságoknak, amely megfelel az $r[i]$ értékeknek.

Részfeladatok

1. (5 pont) $k = 2$
2. (14 pont) $n \leq 5000, 2 \cdot k > n$
3. (13 pont) $2 \cdot k > n$
4. (17 pont) A helyes válasz minden `compare_plants` hívásra 1 vagy -1 .
5. (11 pont) $n \leq 300, q \leq \frac{n \cdot (n-1)}{2}$
6. (15 pont) $x = 0$ minden `compare_plants` hívásra.
7. (25 pont) Nincs további korlátozás.

Minta értékelő

Az alábbi formában olvassa a bemenetet:

- Az 1. sor: $n \ k \ q$
- A 2. sor: $r[0] \ r[1] \ \dots \ r[n-1]$
- A $3 + i$. sorok ($0 \leq i \leq q - 1$): az i . `compare_plants` hívás $x \ y$ paraméterei.

A következő formában írja ki a választ:

- Az $1 + i$. sor ($0 \leq i \leq q - 1$): a válasz az i . `compare_plants` hívásra.