

Comparando Plantas (plants)

Hazel, a botânica, visitou uma exposição especial no Jardim Botânico de Singapura. Nesta exposição, n plantas de **alturas distintas** estão colocadas em círculo. Estas plantas estão numeradas de 0 a $n - 1$ no sentido horário (sentido dos ponteiros do relógio), sendo que a planta $n - 1$ está ao lado da planta 0.

Para cada planta i ($0 \leq i \leq n - 1$), Hazel comparou a planta i com cada uma das $k - 1$ plantas seguintes no sentido horário, e escreveu o número $r[i]$ que indica quantas destas $k - 1$ plantas são mais altas que a planta i . Assim, cada valor $r[i]$ depende da altura relativa de k plantas consecutivas.

Por exemplo, sejam $n = 5$, $k = 3$ e $i = 3$. As próximas $k - 1 = 2$ plantas no sentido horário a partir da planta $i = 3$ seriam a planta 4 e a planta 0. Se a planta 4 fosse mais alta que a planta 3 e a planta 0 fosse mais baixa que a planta 3, então a Hazel iria escrever $r[3] = 1$.

É garantido que os valores $r[i]$ que a Hazel escreveu estão corretos. Logo, existe pelo menos uma configuração de alturas distintas das plantas que é consistente com estes valores.

É-te pedido para comparar as alturas de q pares de plantas. Infelizmente, não tens acesso à exposição. A tua única fonte de informação é o caderno da Hazel com o valor de k e a sequência de valores $r[0], \dots, r[n - 1]$.

Para cada par de plantas distintas x e y que vão ser comparadas, determina qual das três situações seguintes se aplica:

- A planta x é garantidamente mais alta que a planta y : em qualquer configuração de alturas distintas $h[0], \dots, h[n - 1]$ que seja consistente com o array r , temos que $h[x] > h[y]$.
- A planta x é garantidamente mais baixa que a planta y : em qualquer configuração de alturas distintas $h[0], \dots, h[n - 1]$ que seja consistente com o array r , temos que $h[x] < h[y]$.
- A comparação é inconclusiva: nenhum dos casos anteriores se aplica.

Detalhes de implementação

Deves implementar as seguintes funções:

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

- k : o número de plantas consecutivas cujas alturas determinam cada um dos valores de $r[i]$.
- r : um array de tamanho n , onde $r[i]$ é o número de plantas mais altas que a planta i de entre as $k - 1$ plantas seguintes no sentido horário.

- Esta função será chamada exatamente uma vez, antes de qualquer chamada à função `compare_plants`.

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

- x, y : índice das plantas que queremos comparar.
- Esta função deve devolver:
 - 1 se a planta x é garantidamente mais alta que a planta y ,
 - -1 se a planta x é garantidamente mais baixa que a planta y ,
 - 0 se a comparação for inconclusiva.
- Esta função é chamada exatamente q vezes.

Exemplos

Exemplo 1

Considera a seguinte chamada:

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

Imaginemos que o avaliador chama `compare_plants(0, 2)`. Como $r[0] = 0$ podemos imediatamente inferir que a planta 2 não é mais alta que a planta 0. Logo, a chamada deve devolver 1.

Imaginemos que o avaliador chama `compare_plants(1, 2)` a seguir. Para todas as possíveis configurações de alturas que respeitam as restrições acima descritas, a planta 1 é mais baixa que a planta 2. Logo, a chamada deve devolver -1 .

Exemplo 2

Considera a seguinte chamada:

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

Imaginemos que o avaliador chama `compare_plants(0, 3)`. Como $r[3] = 1$, sabemos que a planta 0 é mais alta que a planta 3. Logo, a chamada deve devolver 1.

Imaginemos que o avaliador chama `compare_plants(1, 3)` a seguir. As configurações de alturas $[3, 1, 4, 2]$ e $[3, 2, 4, 1]$ são ambas consistentes com as medições da Hazel. Como a planta 1 é mais baixa que a planta 3 numa das configurações e mais alta que a planta 3 na outra, esta chamada deve devolver 0.

Restrições

- $2 \leq k \leq n \leq 200\,000$
- $1 \leq q \leq 200\,000$
- $0 \leq r[i] \leq k - 1$ (para todo o $0 \leq i \leq n - 1$)
- $0 \leq x < y \leq n - 1$
- Existe uma ou mais configurações de **alturas distintas** das plantas consistentes com o array r .

Subtarefas

1. (5 pontos) $k = 2$
2. (14 pontos) $n \leq 5000$, $2 \cdot k > n$
3. (13 pontos) $2 \cdot k > n$
4. (17 pontos) A resposta correta para cada chamada a `compare_plants` é 1 ou -1 .
5. (11 pontos) $n \leq 300$, $q \leq \frac{n \cdot (n-1)}{2}$
6. (15 pontos) $x = 0$ para cada chamada de `compare_plants`.
7. (25 pontos) Nenhuma restrição adicional.

Avaliador exemplo

O avaliador exemplo lê o input no seguinte formato:

- linha 1: $n \ k \ q$
- linha 2: $r[0] \ r[1] \ \dots \ r[n - 1]$
- linha $3 + i$ ($0 \leq i \leq q - 1$): $x \ y$ para a i -ésima chamada a `compare_plants`

O avaliador exemplo escreve o output no seguinte formato:

- linha $1 + i$ ($0 \leq i \leq q - 1$): o valor devolvido na i -ésima chamada a `compare_plants`.