Jumps

Java May'19 DSA Linear 1 - 1 day 19:28:18

Given a sequence of elements(numbers), calculate the longest possible sequence of 'jumps' from each number.

Each 'jump' must be made according to the following rules:

- You can only 'jump' on a number that is greater than the current one;
- You can 'jump' on a number, only if there isn't one with a greater value between;
- You can 'jump' only from left to right;

Input

Read from the standard input

- On the first line, you will find the number `N`
  - The number of elements
- On the second line you will find `N` numbers, separated by a space
  - The elements themselves

The input will be correct and in the described format, so there is no need to check it explicitly.

Output

Print to the standard output

- On the first line, print the length of the longest sequence of jumps
- On the second line, print the lengths of the sequences, starting from each element

Constraints

- The `N` will always be less than 103344

Sample Tests

Submit solution

My submissions
All submissions
Best submissions

Read editorial

Points: 100 (partial)
Time limit: 0.2s
Java: 1.0s
Memory limit: 64M
Java: 32M
Author: donchominkov

Tags
Linear Data Structures
Difficulty
Intermediate

Allowed languages
java

```
6
1 4 2 6 3 4
```

Output

Copy

```
2
2 1 1 0 1 0
```

Explanation

- Element 1:
  - `1 -> 4 -> 6` (2 jumps)
- Element 2:
  - `4 -> 6` (1 jump)
- Element 3:
  - `2 -> 6` (1 jump)
- Element 4:
  - `6` (0 jumps)
- Element 5:
  - `3 -> 4` (1 jump)
- Element 6:
  - `4` -> (0 jumps)

Input

Java May'19 DSA Linear 1 - 1 day 19:28:18

Output

Copy

```
0
0 0 0 0 0
```

Clarifications

No clarifications have been made at this time.