

Competitive Programming and Contests

12/06/2018

Increasing subarray

You are given an array $A[0..n-1]$ of positive integers. We use $A[i:j]$ to denote the subarray of A of size $j-i+1$, i.e., elements $A[i], A[i+1], \dots, A[j-1], A[j]$.

We say that $A[i:j]$ is *strictly increasing subarray* iff $A[k] < A[k+1]$ for any $i \leq k < j$.

The goal is to report the length of the longest subarray of A such that it is possible to change at most one element to obtain a strictly increasing subarray.

We point out that there exist

1. A $\Theta(n^2)$ time solution. If you find and implement this solution, your grade will be 22;
2. A $\Theta(n)$ time solution. If you find and implement this solution, your grade will be 30.

Input. The first line contains the value of n . The next line consists in n integers, separated by a space.

Output. The number of length of the longest subarray.

Example

Input

```
6          // n
4 2 1 4 5 3 // A
```

Output

4

The subarray 2, 1, 4, 5 is the subarray of that length.

Input

```
5          // n
1 2 3 4 1  // A
```

Output

```
5
The subarray 1, 2, 3, 4, 1 is the subarray of that length.
```