https://course.acciojob.com/idle?question=febaa8a7-8536-4c32-ba4c-f2713863aa35

MEDIUM

Max Score: 40 Points

Max Distance

Given an array `A` of `N` positive elements. Your task is to find maxmimum of `j-i` such that `A[i]<=A[j] and i<=j`.

Input Format

The first line of input contains a single integer, `N`. The second line of input contains `N` space seperated integer representing `position`.

Output Format

Print an integer denoting the maximum value of `j - i`;

Example 1

Input

```
4
3 5 4 2
```

Output

```
2
```

Explanation

Maximum value occurs for pair (3, 4).

Example 2

Input

```
2
1 10
```

Output

```
1
```

Explanation

A[0]<A[1] so (j-i) is 1-0 = 1.

Constraints

N <= N <= 10^5

1 <= A[i] <= 10^9

Topic Tags

- **2-Pointers**
- **Greedy**
- **Binary Search**
- **Arrays**

My code

```
// in java
import java.io.*;
import java.util.*;

class Main {
    public static void main (String[] args) throws IOException {
        BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));

          int n = Integer.parseInt(br.readLine().trim()); // size of
array
          int arr[] = new int[n];
          String inputLine[] = br.readLine().trim().split(" ");
          for(int i=0; i<n; i++){
            arr[i] = Integer.parseInt(inputLine[i]); // input
elements of array
          }

          Solution ob = new Solution();

          System.out.println(ob.maxIndexDiff(arr, n)); // print the
result

    }
}


class Solution{


  static int maxIndexDiff(int A[], int N) {

    // Your code here
      int n=N;
      int max=0;
        int arr[]=new int[n];//containg right to left asending order
        // pic n-1 element if grater find then take it flag and put in
arr of this flag
        int flag=A[n-1];
      for(int i=n-1;i>=0;i--)
                {
                    if(A[i]>flag)
                        flag =A[i];
                    arr[i]=flag;
                }
                //now use binary seartch for j pointer
        int ans=0;
                  for(int i=0; i<n; i++)
                        {
                        int lp=i,rp=n-1;

                            while(lp<=rp)
                                {
                                    int m=(lp+rp)/2;
                                    if(arr[m]>=A[i])
                                    {
                                        int l=m-i;
                                        if(ans<l)
                                            ans=l;
                                        lp=m+1;
                                    }
                                    else rp=m-1;
                                }
                        }
        return ans;
    }
}
```