2863. Maximum Length of Semi-Decreasing Subarrays

Description

You are given an integer array `nums`.

Return *the length of the* **longest semi-decreasing** *subarray of* `nums` *, and* `0` *if there are no such subarrays.*

- A **subarray** is a contiguous non-empty sequence of elements within an array.
- A non-empty array is **semi-decreasing** if its first element is **strictly greater** than its last element.

Example 1:

```
Input: nums = [7,6,5,4,3,2,1,6,10,11]
Output: 8
Explanation: Take the subarray [7,6,5,4,3,2,1,6].
The first element is 7 and the last one is 6 so the condition is met.
Hence, the answer would be the length of the subarray or 8.
It can be shown that there aren't any subarrays with the given condition with a length greater than 8.
```

Example 2:

```
Input: nums = [57,55,50,60,61,58,63,59,64,60,63]
Output: 6
Explanation: Take the subarray [61,58,63,59,64,60].
The first element is 61 and the last one is 60 so the condition is met.
Hence, the answer would be the length of the subarray or 6.
It can be shown that there aren't any subarrays with the given condition with a length greater than 6.
```

Example 3:

```
Input: nums = [1,2,3,4]
Output: 0
Explanation: Since there are no semi-decreasing subarrays in the given array, the answer is 0.
```

Constraints:

- `1 <= nums.length <= 10^5`
- `-10^9 <= nums[i] <= 10^9`