Let's call any (contiguous) subarray B (of A) a *mountain* if the following properties hold:

* B.length >= 3
* There exists some 0 < i < B.length - 1 such that B[0] < B[1] < ... B[i-1] < B[i] > B[i+1] > ... > B[B.length - 1]

(Note that B could be any subarray of A, including the entire array A.)

Given an array A of integers, return the length of the longest *mountain*.

Return 0 if there is no mountain.

**Example 1:**

**Input:** [2,1,4,7,3,2,5]

**Output:** 5

**Explanation:** The largest mountain is [1,4,7,3,2] which has length 5.

**Example 2:**

**Input:** [2,2,2]

**Output:** 0

**Explanation:** There is no mountain.

**Note:**

1. 0 <= A.length <= 10000
2. 0 <= A[i] <= 10000

**Follow up:**

* Can you solve it using only one pass?
* Can you solve it in O(1) space?