Given an array nums sorted in **non-decreasing** order, return *the maximum between the number of positive integers and the number of negative integers.*

* In other words, if the number of positive integers in nums is pos and the number of negative integers is neg, then return the maximum of pos and neg.

**Note** that 0 is neither positive nor negative.

**Example 1:**

Input: nums = [-2,-1,-1,1,2,3]  
Output: 3  
Explanation: There are 3 positive integers and 3 negative integers. The maximum count among them is 3.

**Example 2:**

Input: nums = [-3,-2,-1,0,0,1,2]  
Output: 3  
Explanation: There are 2 positive integers and 3 negative integers. The maximum count among them is 3.

**Example 3:**

Input: nums = [5,20,66,1314]  
Output: 4  
Explanation: There are 4 positive integers and 0 negative integers. The maximum count among them is 4.

**Constraints:**

* 1 <= nums.length <= 2000
* -2000 <= nums[i] <= 2000
* nums is sorted in a **non-decreasing order**.

**Follow up:** Can you solve the problem in O(log(n)) time complexity?