Problem 1150: Check If a Number Is Majority Element in a Sorted Array

Problem Information

Difficulty: Easy
Acceptance Rate: 59.88%
Paid Only: Yes
Tags: Array, Binary Search

Problem Description

Given an integer array `nums` sorted in non-decreasing order and an integer `target`, return `true` _if_ `target` _is a**majority** element, or _`false` _otherwise_.

A **majority** element in an array `nums` is an element that appears more than `nums.length / 2` times in the array.

Example 1:

Input: nums = [2,4,5,5,5,5,5,6,6], target = 5 **Output:** true **Explanation:** The value 5 appears 5 times and the length of the array is 9. Thus, 5 is a majority element because 5 > 9/2 is true.

Example 2:

Input: nums = [10,100,101,101], target = 101 **Output:** false **Explanation:** The value 101 appears 2 times and the length of the array is 4. Thus, 101 is not a majority element because 2 > 4/2 is false.

Constraints:

* `1 <= nums.length <= 1000` * `1 <= nums[i], target <= 109` * `nums` is sorted in non-decreasing order.

Code Snippets

C++:

```
class Solution {
public:
bool isMajorityElement(vector<int>& nums, int target) {

}
};
```

Java:

```
class Solution {
public boolean isMajorityElement(int[] nums, int target) {

}
}
```

Python3:

```
class Solution:
def isMajorityElement(self, nums: List[int], target: int) -> bool:
```