169. Majority Element

Easy 9249 331 Add to List Share

Given an array `nums` of size `n`, return *the majority element*.

The majority element is the element that appears more than `⌊n / 2⌋` times. You may assume that the majority element always exists in the array.

Example 1:

```
Input: nums = [3,2,3]
Output: 3
```

Example 2:

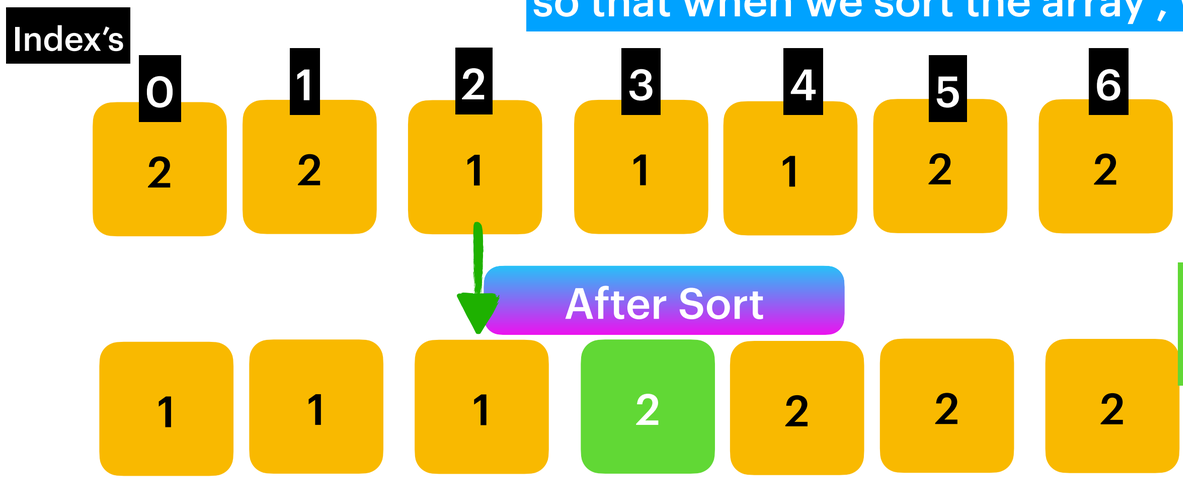
```
Input: nums = [2,2,1,1,1,2,2]
Output: 2
```

Constraints:

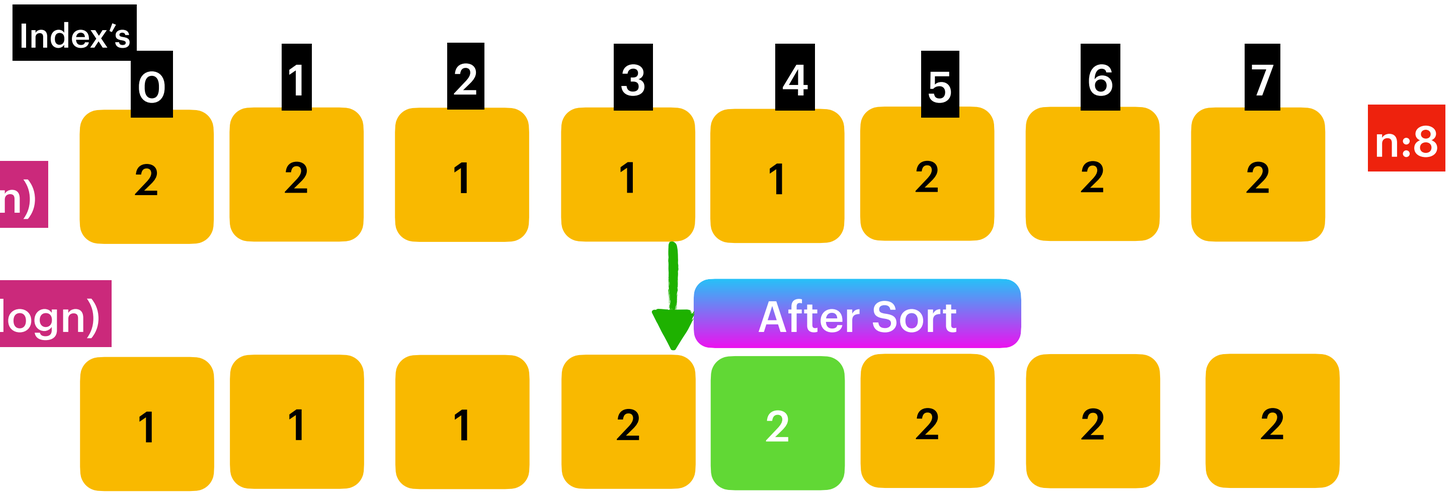
- `n == nums.length`
- `1 <= n <= 5 * 10^4`
- `-10^9 <= nums[i] <= 10^9`

169 Majority Element

Key is Majority Element repeats > n/2 times
so that when we sort the array , we can always find Majority element in n/2 index.



Majority Element is on n/2 index :
nums[3] = 2



Time Complexity : O(nlogn)

Merge Sort O(n)

Sorting

Space Complexity : O(1)

Quick Sort O(logn)

Can we improve the Time&Space Complexity ?

Yes By applying Math.

Majority Element is on n/2 index :
nums[4] = 2

Lets apply voting:

As we know that within a given input the majority element presents > n/2 times.

When we apply voting alway
we left with the Majority Element.

When the element repeats increment the vote, other time decrement the vote.

When vote becomes zero, take the current integration element Takes the vote.

Let's Apply Math

Time Complexity : O(n)

Space Complexity : O(1)

Return the Element i.e 2

4. Median of Two Sorted Arrays

Hard 15850 1951 Add to List Share

Given two sorted arrays `nums1` and `nums2` of size `m` and `n` respectively, return **the median** of the two sorted arrays.

The overall run time complexity should be `O(log (m+n))`.

Example 1:

```
Input: nums1 = [1,3], nums2 = [2]
Output: 2.00000
Explanation: merged array = [1,2,3] and median is 2.
```

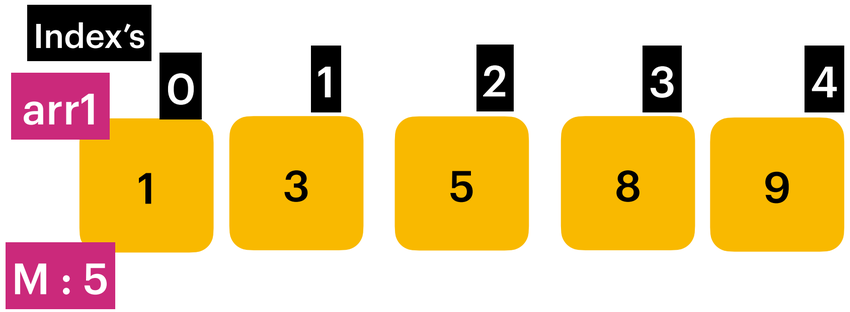
Example 2:

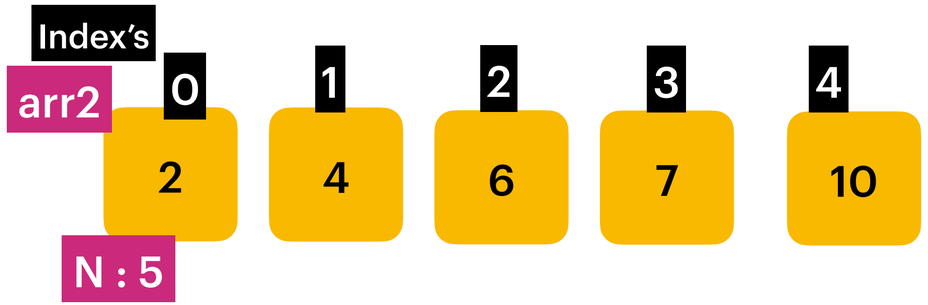
```
Input: nums1 = [1,2], nums2 = [3,4]
Output: 2.50000
Explanation: merged array = [1,2,3,4] and median is (2 + 3)
/ 2 = 2.5.
```

Constraints:

- `nums1.length == m`
- `nums2.length == n`
- `0 <= m <= 1000`
- `0 <= n <= 1000`
- `1 <= m + n <= 2000`
- `-10^6 <= nums1[i], nums2[i] <= 10^6`

4 Median Of Two Sorted Arrays



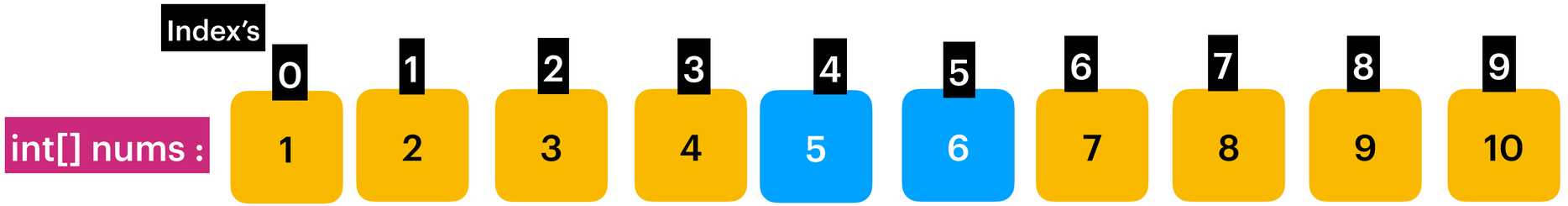


Brute Force

Take an Array Of size M+N , combine and merge both the arrays.

If the length is even then (nums[mid] + nums[mid-1] / 2) is the Median.

If the length is odd then nums[mid] is the Median.



As the length is even then (nums[mid] + nums[mid-1] / 2) is the Median.

= (5 + 6) / 2 = 11

Time Complexity : O(M+N)

Space Complexity : O(M+N)

Can we improve the Time&Space Complexity ?

Yes By applying Math.