HW 12: Majority Element (Leetcode-169)

PROBLEM STATEMENT:

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.

☑ISKA MTLB YEH HAI: majority element always equal to (mid) index par exist krega jab array sorted form me hogaa;

Example 1:

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

Example 2:

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

Output: 2

BRUTE FORCE APPROACH 01:

Step 01: count each elements Step 02: compare each elements

Step 03: return highest counted element = majority element

Time Complexity: $O(N^2)$, where N is Length of array

Space Complexity: O(1), no extra space used

✓BINARY SEARCH APPROACH 02:

Step 01: first occurrence of mid Step 02: last occurrence of mid

Step 03: total occurrence of mid = majority element

Time Complexity: O(N Log N), where N is Length of array

Space Complexity: O(1), no extra space used

SORTING APPROACH 03:

Step 01: sort the array

Step 02: return the mid index element = majority element

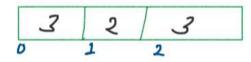
Time Complexity: O(N Log N), where N is Length of array

Space Complexity: O(1), no extra space used



Example 1

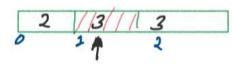
Input: nums = [3,2,3]



BRUTE FORCE APPROACH 01:

Out Put: 3

✓SORTING APPROACH 03:



Always mid Index
ELAMINT 95 Majority
ELAMINT = 3

✓BINARY SEARCH APPROACH 02:

SORT

$$Mid = \frac{0+2}{2} = 1$$

Last occ. of mid = 2 First occ. of mid = 1 Total occ. of mid = 2-1+1 = 2

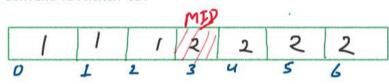
Input: nums = [2, 2, 1, 1, 1, 2, 2]

2	2	1	1	1	2	2	
0	1	2	3	ч	5	6	

BRUTE FORCE APPROACH 01:

Output = 2

SORTING APPROACH 03:



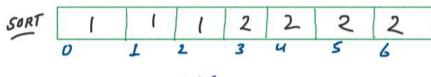
Always mid Index

ELLAUNT 95 majority

ELLAUNT = 2

Of mid Induk

BINARY SEARCH APPROACH 02:



$$mid = \frac{0+6}{2} = 3$$

mid Indux Elamin + = NUMSE mid]

Last occ. of mid = 6 First occ. of mid = 3 Total occ-of mid = 6-3+1= 1 \Rightarrow Return Nums[4] = 2 Output

SORTING APPROACH 03: PROGRAM

```
// HW 12: Majority Element (Leetcode-169)
// SORTING APPROACH 03:
class Solution {
public:
    int majorityElement(vector<int>& nums) {
        sort(nums.begin(), nums.end());
        int midIndex = (0 + nums.size()-1)/2;
        return nums[midIndex];
    }
};
```