Majority Element -1

169. Majority Element

Solved

Easy | Topics | Companies

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

$1 <= nums.length <= 5 \times 10^4$

$-10^9 <= nums[i] <= 10^9$] so can use int

arr = 1 1 2 3 4 1 1 6 1 1
0 1 2 3 4 5 6 7 8 9

n=10 here $\frac{n}{2} = 5$

freq[1] = 6 > 5 so 1 is Majority Element

Brute force → Use HashMap & Count frequency & which Even freq > n/2 then that is majority Element

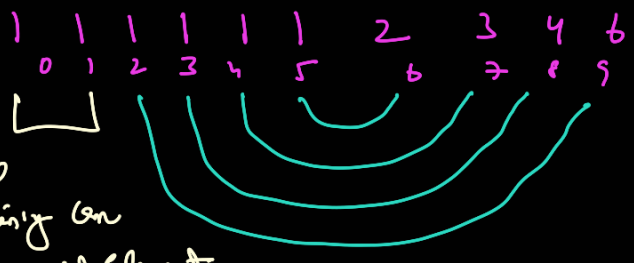
TC → O(n)

SC → O(n)

Actual soln [Moore Voting Algo] SC → O(1) TC → O(n)

arr = 1 1 2 3 4 1 1 6 1 1
0 1 2 3 4 5 6 7 8 9

pairing the like →



pair distinct elements

Now Remaining can be Majority Element

arr1 = 1 2 3 4 5 6 6 7 9 9

Cant pair as same

So we need to do pair of distinct Element

Distinct Element the count -- & same then ++ & jaun ki count 0 then put in dist of possible Majority Element

1 2 3 4 5 6 6 7 9 9
0 1 2 3 4 5 6 7 8 9

val = arr[0] count = 1 initially

i = 1 to n

```
if (val == arr[i]) count++;
else {
    count--;
    if (count == 0) { val = arr[i]; count = 1; }
}
```

1 1 2 3 4 5
0 1 2 3 4 5

Val = 1 Count = 1

i = 1 → 1 == 1 so count++ → val = 1 count = 2 → 1 1 2

i = 2 → 1 != 2 count-- → val = 1 count = 1 → 1 1 2 3

i = 3 → 1 != 3 Count-- → val = 1 count = 0 → 1 1 2 3

As count = 0 so Abb update val = 3 count = 1 Abb 3 se matching honge

i = 4 → 3 != 4 Count-- → val = 3 count = 0

so update val = 4 count = 1

i = 5 → 5 != 4 Count-- → val = 4 count = 0

update val = 5 count = 1

So 5 can be our majority element

Code | C++

```
class Solution {
public:
    int majorityElement(vector<int>& nums) {
        int n=nums.size();
        int count=1;
        int v=nums[0];
        for(int i=1;i<n;i++){
            if(nums[i]==v) count++;
            else{
                count--;
                if(count==0){
                    count=1;
                    v=nums[i];
                }
            }
        }
        return v;

    }
};
```

View less

yhe Question bolta hai hi element Exist krta hai isliye just need to return last value

But if majority element doesn't Exist we need to verify this ki last val majority hai ya nhi !!

To verify this v is majority or not

Count freq of v in Array Again

if (freq_v >= n/2) return v;

else return -1;