

### 33. Majority Element II | Brute-Better-Optimal

Given an integer array of size  $n$ , find all elements that appear more than  $\lfloor n/3 \rfloor$  times.

**Example 1:**

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

**Example 2:**

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

**Example 3:**

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

**Constraints:**

- $1 \leq \text{nums.length} \leq 5 * 10^4$
- $-10^9 \leq \text{nums}[i] \leq 10^9$

$$\text{arr} \mathcal{E} 7 = \{1, 1, 1, 3, 3, 2, 2, 2\} \quad n=8$$

$$\text{Or} \Rightarrow \{1, 2\}$$

$$8/3 = 2.5 \approx 2$$

return the integer appearing more than  $\geq 2$ .

Q) How many integers at max be there in the answer?

$\Rightarrow$  At max is '2' element.

At min is '0' element.

Brute force :

1) Pick up every element and check through the array.

more than floor of  $n/3$  times

2) Take a list which is empty

3) Pick up 'i' and traverse through the array and count the no of it.

{1, 1, 1, 3, 3, 2, 2, 2}

↖ ↖ ↖ ↗

①  $\Rightarrow s > \lceil n/3 \rceil$

list {1, 2}

②  $\Rightarrow s > \lceil n/3 \rceil$

If the list is '2' element  
then stop it.

ls = []

for ( $i=0 \rightarrow n-1$ )

{  
if ( $(ls.size() == 0) || ls[0] != num[i]$ )

↳

cnt = 0;

for ( $j=0 \rightarrow n-1$ )

if ( $num[j] == num[i]$ )

cnt++;

}

y

if (cnt > n/3)

ls.add (nums[i]),

if (ls.size() == 0) break

T.C  $\Rightarrow \Theta(n^2)$

S.C  $\Rightarrow \Theta(2)$

Better Solutions :

{1, 1, 1, 1, 3, 3, 2, 2, 2}    n = 8  
↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑

using hashing.

[1, 2]

{  
  (2, 1)  $\geq 3$   
  (3, 1)  $\geq 2$   
  (1, 1)  $\geq 4$   
}

→ Iterate over the array and store

it in the hashmap and increase

the cnt if the value is same

(no, cnt)

→ Iterate over the hash map and return the list  $\geq n/3$  times.

with single iteration

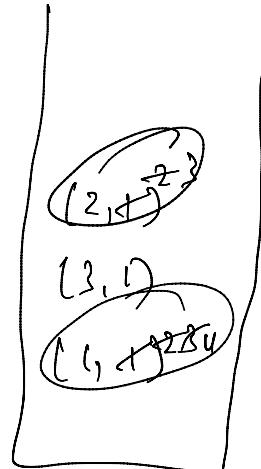
$$\lceil \frac{n}{3} \rceil = 2$$

so we need an element which should appear more than '2' times

so minimum = 3.

Iterate over the array and store it in the hashmap

if the count is '3' return it to the list



T.C  $\Rightarrow O(n)$  &

IS = [] map  $mm = \lceil \frac{n}{3} \rceil + 1$

for ( $i = 0 \rightarrow n - 1$ )

(3, 1)  
(2, +)  $\cancel{+}$

{  $mpp[\text{arr}[i]]++;$   $\rightarrow (1, +) \cancel{+} \cancel{+}$   
no , count

if ( $mpp[\text{arr}[i]] == mm$ )

ls.add( $\text{arr}[i]$ )

}

T.C  $\Rightarrow O(n) \times \log n$  S.C  $\Rightarrow O(n)$

```

i C++ Autocomplete
1 class Solution {
2 public:
3     vector<int> majorityElement(vector<int>& v) {
4         vector<int> ls;
5         map<int, int> mpp;
6         int n = v.size();
7         int mini = (int)(n/3) + 1;
8         for(int i = 0; i < n; i++) {
9             mpp[v[i]]++;
10            if(mpp[v[i]] == mini) {
11                ls.push_back(v[i]);
12            }
13            if(ls.size() == 2) break;
14        }
15        sort(ls.begin(), ls.end());
16        return ls;
17    }
18}

```

### 3) Optimal Solution

$$\text{arr}[] = \{1, 1, 1, 1, 3, 2, 2, 2\} \quad n=8$$

$\frac{n}{2}$  times

cnt = 0, el

for(i=0 → i++)

{ if (cnt == 0)

    cnt = 1

    el = arr[i];

Replicate  
but it won't  
work

$\frac{n}{3}$  times

cnt1 = 0, cnt2 = 0, el1, el2

for (i=0 → i++)

{ if (cnt1 == 0 & arr[i] != el2)

    cnt1 = 1, el1 = arr[i]

} else if (cnt2 == 0 & arr[i] != el1)

{ . . . 7

$e1 = arr[i];$  — but it won't work  
 $\{ \text{else if } (e1 == arr[i]) \quad \text{work} \}$   
 $\{ \text{cnt}_2 = 1, e1_2 = arr[i] \}$   
 $\}$   
 $\text{cnt}++; \quad \text{for this last case}$   
 $\text{else if } (e1_1 == arr[i])$   
 $\text{else}$   
 $\text{cnt}--; \quad [2, 1, 1, 3, 1, 4, 5, 6]$   
 $\text{cnt}_1++;$   
 $\text{else if } (e1_2 == arr[i])$   
 $\text{cnt}_2++$   
 $\}$   
 $\text{else}$   
 $\text{cnt}_1--, \text{cnt}_2--$

Initiation for  $n/2$  time : It was cancellation logic

Same for  $n/3$  times (Watch  $n/2$  time video).

```

i C++ Autocomplete
2 public:
3     vector<int> majorityElement(vector<int>& v) {
4         int cnt1 = 0, cnt2 = 0;
5         int el1 = INT_MIN;
6         int el2 = INT_MIN;
7
8         for(int i = 0; i < v.size(); i++){
9             if(cnt1 == 0 && el2 != v[i]){
10                 cnt1 = 1;
11                 el1 = v[i];
12             }
13             else if (cnt2 == 0 && el1 != v[i]){
14                 cnt2 = 1;
15                 el2 = v[i];
16             }
17             else if(v[i] == el1) cnt1++;
18             else if(v[i] == el2) cnt2++;
19             else {
20                 cnt1--, cnt2--;
21             }
22         }
23
24         vector<int> ls;
25         cnt1 = 0, cnt2 = 0;
26         for(int i = 0; i < v.size(); i++){
27             if(el1 == v[i]) cnt1++;
28             if(el2 == v[i]) cnt2++;
29         }
30         int mini = (int)(v.size()/3) + 1;
31         if(cnt1 >= mini) ls.push_back(el1);
32         if(cnt2 >= mini) ls.push_back(el2);
33         sort(ls.begin(), ls.end());
34         return ls;
35     }
36

```

$T.C \Rightarrow O(2^n)$        $S.C \Rightarrow O(1)$