

229. Majority Element II

Add to List ▾

Question

Editorial Solution

My Submissions (/problems/majority-element-ii/submissions/)

Total Accepted: **45851** Total Submissions: **165066** Difficulty: **Medium** Contributors: **Admin**

Given an integer array of size n , find all elements that appear more than $\lfloor n/3 \rfloor$ times. The algorithm should run in linear time and in $O(1)$ space.

Show Hint ►

Subscribe (/subscribe/) to see which companies asked this question

Show Tags

Show Similar Problems

Have you met this question in a real interview? ☐ Yes ☐ NoDiscuss (<https://discuss.leetcode.com/category/237>)

Top Solutions

Pick One (/problems/random-one-question/)

Python ▾



```
1 class Solution(object):
2     def majorityElement(self, nums):
3         """
4         :type nums: List[int]
5         :rtype: List[int]
6         """
7         from collections import Counter
8         cnt= Counter(nums).most_common()
9         m=[]
10        for key,value in cnt:
11            if value>math.floor(len(nums)/3):
12                m.append(key)
13        return m
```

Notes

Custom Testcase ☐

Contribute Testcase ?

Run Code

Submit Solution

✉ Send Feedback (<mailto:admin@leetcode.com?subject=Feedback>)