

# 169 Majority Element

2018年4月3日 17:21

## Question:

Given an array of size  $n$ , find the majority element. The majority element is the element that appears **more than**  $\lfloor n/2 \rfloor$  times.

You may assume that the array is non-empty and the majority element always exist in the array.

来自 <<https://leetcode.com/problems/majority-element/description/>>

给定一个大小为  $n$  的数组，找到其中的众数。众数是指在数组中出现次数大于  $\lfloor n/2 \rfloor$  的元素。你可以假设数组是非空的，并且数组中的众数永远存在。

## Solution for Python3:

```
1  class Solution:
2      def majorityElement(self, nums):
3          """
4              :type nums: List[int]
5              :rtype: int
6          """
7          c = 1
8          k = nums[0]
9          for i in range(1, len(nums)):
10             if nums[i] == k:
11                 c += 1
12             else:
13                 if c:
14                     c -= 1
15                 else:
16                     c = 1
17                     k = nums[i]
18             return k
```

## Solution for C++:

```
1  class Solution {
2  public:
3      int majorityElement(vector<int>& nums) {
4          int k = nums[0], c = 1;
5          for (int i = 0; i < nums.size(); i++) {
6              if (nums[i] == k) {
7                  c++;
8              } else {
9                  if (c > 1) {
10                     c--;
11                 } else {
12                     k = nums[i];
13                 }
14             }
15         }
16         return k;
17     }
```

