

# 面试题39：数组中出现次数超过一半的数字

## 题目描述

数组中有一个数字出现的次数超过数组长度的一半，请找出这个数字。例如输入一个长度为9的数组{1,2,3,2,2,2,5,4,2}。由于数字2在数组中出现了5次，超过数组长度的一半，因此输出2。如果不存在则输出0。

## 解题思路

### ❖ 解法一：基于 Partition 函数的时间复杂度为 $O(n)$ 的算法

如果我们回到题目本身仔细分析，就会发现前面的思路并没有考虑到数组的特性：数组中有一个数字出现的次数超过了数组长度的一半。如果把这个数组排序，那么排序之后位于数组中间的数字一定就是那个出现次数超过数组长度一半的数字。也就是说，这个数字就是统计学上的中位数，即长度为  $n$  的数组中第  $n/2$  大的数字。我们有成熟的时间复杂度为  $O(n)$  的算法得到数组中任意第  $k$  大的数字。

## ❖ 解法二：根据数组特点找出时间复杂度为 $O(n)$ 的算法

接下来我们从另外一个角度来解决这个问题。数组中有一个数字出现的次数超过数组长度的一半，也就是说它出现的次数比其他所有数字出现次数的和还要多。因此，我们可以考虑在遍历数组的时候保存两个值：一个是数组中的一个数字；另一个是次数。当我们遍历到下一个数字的时候，如果下一个数字和我们之前保存的数字相同，则次数加 1；如果下一个数字和我们之前保存的数字不同，则次数减 1。如果次数为零，那么我们需要保存下一个数字，并把次数设为 1。由于我们要找的数字出现的次数比其他所有数字出现的次数之和还要多，那么要找的数字肯定是最后一次把次数设为 1 时对应的数字。

## C++

```
1 class Solution {
2 public:
3     int MoreThanHalfNum_Solution(vector<int>
numbers) {
4         int n = numbers.size();
5         if (n == 0) return 0;
6         int num = numbers[0], count = 1;
7         for (int i = 1; i < n; i++) {
8             if (numbers[i] == num) count++;
9             else count--;
10            if (count == 0) {
11                num = numbers[i];
12                count = 1;
13            }
14        }
```

```
15 // verifying
16 count = 0;
17 for (int i = 0; i < n; i++) {
18     if (numbers[i] == num) count++;
19 }
20 if (count * 2 > n) return num;
21 return 0;
22 }
23 };
```

## Java

```
1 public class solution {
2     public int MoreThanHalfNum_Solution(int
3     [] array) {
4         if(array == null || array.length <=
5         0)
6             return 0;
7         int result = array[0];
8         int times = 1;
9         for(int i = 1; i < array.length;
10        i++){
11             if(times == 0){
12                 result = array[i];
13                 times = 1;
14             }
15             else if(array[i] == result)
16                 times++;
17         }
18     }
19 }
```

```
14         else {
15             times--;
16         }
17     }
18     int count = 0;
19     for(int i = 0; i<array.length; i++){
20         if(array[i] == result)
21             count++;
22     }
23     if(count * 2 <= array.length)
24         return 0;
25     return result;
26 }
27 }
```

## Python 2.7.3

```
1 # -*- coding:utf-8 -*-
2 class Solution:
3     def MoreThanHalfNum_Solution(self,
4     numbers):
5         # write code here
6         dict = {}
7         for no in numbers:
8             if not dict.has_key(no):
9                 dict[no] = 1
10            else:
11                dict[no] = dict[no] + 1
12            if dict[no] > len(numbers)/2:
13                return no
14        return 0
```