# 数组中出现次数超过一半的数字

## 题目

[牛客网](https://www.nowcoder.com/practice/e8a1b01a2df14cb2b228b30ee6a92163?tpId=13&tqId=11181&rp=1&ru=%2Fta%2Fcoding-interviews&qru=%2Fta%2Fcoding-interviews%2Fquestion-ranking&tPage=2)

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

1. 由于数组的特性，在排序数组中，超过半数的数字一定包含中位数
2. 通过 partition 方法，借用快排的思想，随机选取一个 key，将数组中小于 key 的移动到 key 的左侧，数组中大于 key 的移动到 key 的右侧
3. 最终找到中位数的下标，还需要检查中位数是否超过半数

public int MoreThanHalfNum\_Solution(int[] array) {

 int start = 0, end = array.length - 1;
 int mid = array.length / 2;

 int index = partition(array, start, end);
 if (index == mid) {
 return array[index];
 }

 while (index != mid && start <= end) {
 if (index > mid) {
 end = index - 1;
 index = partition(array, start, end);
 } else {
 start = index + 1;
 index = partition(array, start, end);
 }
 }

 if (checkIsHalf(array, index)) return array[index];

 return 0;
}

private boolean checkIsHalf(int[] array, int index) {
 if (index < 0) {
 return false;
 }

 int count = 0;
 for (int i : array) {
 if (array[index] == i) {
 count++;
 }
 }

 return count > array.length / 2;
}

private int partition(int[] array, int start, int end) {
 if (start >= array.length || start < 0
 || end >= array.length || end < 0) {
 return -1;
 }

 int key = array[start];
 int left = start, right = end;

 while (left < right) {
 while (left < right && array[right] >= key) {
 right--;
 }
 if (left < right) {
 array[left] = array[right];
 left++;
 }

 while (left < right && array[left] <= key) {
 left++;
 }
 if (left < right) {
 array[right] = array[left];
 right--;
 }
 }
 array[left] = key;

 return left;
}