**难度: 简单**

给定一个数组 nums，编写一个函数将所有 0 移动到数组的末尾，同时保持非零元素的相对顺序。

示例:

输入: [0,1,0,3,12]

输出: [1,3,12,0,0]

说明:

必须在原数组上操作，不能拷贝额外的数组。

尽量减少操作次数。

来源：力扣（LeetCode）

链接：https://leetcode-cn.com/problems/move-zeroes

著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。

**思路1:**

1). 判断数组的值是否是0,

2). 如果是0,就在找到数组的值不是0的进行互换

3). 如果不是0,两个的下标都加1

|  |
| --- |
| public void moveZeroes(int[] nums) {  for (int i = 0, j = 0; i < nums.length; i++) {  if (nums[i] == 0) {  //循环查找不是0的值  while(nums[j] == 0 && j < nums.length -1){  j++;  }    //互换  int temp = nums[i];  nums[i] = nums[j];  nums[j] = temp;  }else {  j++;  }  }  } |

**思路2:**

1). 让右指针去寻找不等于0的值,然后进行替换

2). 替换之后左指针+1

3). 每次右指针都要持续+1

|  |
| --- |
| public void moveZeroes2(int[] nums) {  //定义左右指针  int l = 0, r = 0;  //循环判断  while(r < nums.length) {  if (nums[r] != 0) {  int temp = nums[l];  nums[l] = nums[r];  nums[r] = temp;  l++;  }  r++;  }  } |