75. Sort Colors(Medium)

Given an array with *n* objects colored red, white or blue, sort them [**in-place**](https://en.wikipedia.org/wiki/In-place_algorithm)so that objects of the same color are adjacent, with the colors in the order red, white and blue.

Here, we will use the integers 0, 1, and 2 to represent the color red, white, and blue respectively.

**Note:** You are not suppose to use the library's sort function for this problem.

**Example:**

**Input:** [2,0,2,1,1,0]

**Output:** [0,0,1,1,2,2]

思路：3个挡板，4块区域，同向而行/反向而行。主pivot为中间量，利用swap，进行比较，小于则放前边并++，等于则直接++，大于则放后边并--。**但这只是3个颜色的情况。**

class Solution {

public void sortColors(int[] nums) {

if(nums == null || nums.length <= 1){

return;

}

int zero = 0;

int one = 0;

int two = nums.length - 1;

while(one <= two){

if(nums[one] == 0){

swap(nums,zero++,one++);

}else if(nums[one] == 1){

one++;

}else{

swap(nums,one,two--);

}

}

}

private void swap(int[] nums,int a, int b){

int tmp = nums[a];

nums[a] = nums[b];

nums[b] = tmp;

}

}