148. Sort Colors

* [Description](http://www.lintcode.com/en/problem/sort-colors/" \l "description)
* [Notes](http://www.lintcode.com/en/problem/sort-colors/#note)
* [Testcase](http://www.lintcode.com/en/problem/sort-colors/#testcase)
* [Judge](http://www.lintcode.com/en/problem/sort-colors/#judge)

Given an array with *n*objects colored *red*, *white* or *blue*, sort them 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.

 Notice

You are not suppose to use the library's sort function for this problem.   
You should do it in-place (sort numbers in the original array).

Have you met this question in a real interview?

Yes

**Example**

Given [1, 0, 1, 2], sort it in-place to [0, 1, 1, 2].

<http://www.lintcode.com/en/problem/sort-colors/#>

/\* This function takes last element as pivot,

places the pivot element at its correct

position in sorted array, and places all

smaller (smaller than pivot) to left of

pivot and all greater elements to right

of pivot \*/

static int partition(int arr[], int low, int high)

{

int pivot = arr[high];

int i = (low-1); // index of smaller element

for (int j=low; j<high; j++)

{

// If current element is smaller than or

// equal to pivot

if (arr[j] <= pivot)

{

i++;

// swap arr[i] and arr[j]

int temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

// swap arr[i+1] and arr[high] (or pivot)

int temp = arr[i+1];

arr[i+1] = arr[high];

arr[high] = temp;

return i+1;

}

/\* The main function that implements QuickSort()

arr[] --> Array to be sorted,

low --> Starting index,

high --> Ending index \*/

static void sort(int arr[], int low, int high)

{

if (low < high)

{

/\* pi is partitioning index, arr[pi] is

now at right place \*/

int pi = partition(arr, low, high);

// Recursively sort elements before

// partition and after partition

sort(arr, low, pi-1);

sort(arr, pi+1, high);

}

}

public static void sortColors(int[] nums) {

// write your code here

sort(nums,0,nums.length-1);

}