

LintCode领扣题解 (/problem) / 颜色分类 · Sort Colors

颜色分类 · Sort Colors

中文

Pocket Gems (/problem/?tags=pocket-gems)

微软 (/problem/?tags=microsoft)

脸书 (/problem/?tags=facebook)

两根指针 (/problem/?tags=two-pointers)

数组 (/problem/?tags=array)

排序 (/problem/?tags=sort)

描述

给定一个包含红, 白, 蓝且长度为 n 的数组, 将数组元素进行分类使相同颜色的元素相邻, 并按照红、白、蓝的顺序进行排序。

我们可以使用整数 0, 1 和 2 分别代表红, 白, 蓝。

❗ 不能使用代码库中的排序函数来解决这个问题。排序需要在原数组中进行。

样例

样例 1

输入 : [1, 0, 1, 2]
输出 : [0, 1, 1, 2]
解释 : 原地排序。

挑战

一个相当直接的解决方案是使用计数排序扫描2遍的算法。

首先, 迭代数组计算 0,1,2 出现的次数, 然后依次用 0,1,2 出现的次数去覆盖数组。

你是否能想出一个仅使用常数级额外空间复杂度且只扫描遍历一遍数组的算法?

在线评测地址: <https://www.lintcode.com/problem/sort-colors/> (<https://www.lintcode.com/problem/sort-colors/>)

收起题目描述 ^

语言类型

ALL (29)

python (11)

java (10)

cpp (6)

javascript (2)

上传题解



令狐冲

更新于 11/6/2020, 3:30:55 PM

使用一次扫描的办法。设立三根指针, left, index, right。定义如下规则:

- left 的左侧都是 0 (不含 left)
- right 的右侧都是 2 (不含 right)

index 从左到右扫描每个数, 如果碰到 0 就丢给 left, 碰到 2 就丢给 right。碰到 1 就跳过不管。

```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm\_source=code
 */
class Solution:
    """
    @param nums: A list of integer which is 0, 1 or 2
    @return: nothing
    """
    def sortColors(self, A):
        left, index, right = 0, 0, len(A) - 1
        # be careful, index < right is not correct
        while index <= right:
            if A[index] == 0:
                A[left], A[index] = A[index], A[left]
                left += 1
                index += 1 # move to next number
            elif A[index] == 2:
                A[right], A[index] = A[index], A[right]
                right -= 1
            else: # == 1, skip
                index += 1
```

👍 获赞 19

💬 2 条评论

你的口袋题库

2000+ 算法真题、国内外名企题库免费开放



九章算法APP

九章-小原

更新于 11/24/2020, 6:59:01 AM

解题思路

- 最直观的方式是对数组进行排序, 如快速排序, 时间复杂度是 $O(n \log n)$, 但这种做法没有利用元素值在有限范围内的特性。
- 还有种方法是计数排序, 先扫描一遍记录0、1和2的个数, 然后再扫描一遍对数组进行赋值。时间复杂度是 $O(n)$, 比快排快了不少, 但是依旧没有利用元素只有3种的特性。
- 我们这里采用双指针的方法, 一遍扫描即可。时间复杂度为 $O(n)$, 只需常量空间。

算法流程

- 我们建立首尾双指针 `left` 和 `right`, 分别指示0/1边界和1/2边界。 `left` 左边 (不含 `left`) 全为0, `right` 右边 (不含 `right`) 全为2。初始化 `left` 和 `mid` 为0, `right` 为 `len(nums)-1`。
- 第三个指针 `mid` 从 `left` 起向 `right` 移动, 边扫描边实时更新两个边界。
 - 若 `nums[mid]` 为0: 交换第 `mid` 个和第 `left` 个元素, 并将 `left` 指针和 `mid` 指针都向右移。
 - 若 `nums[mid]` 为2: 交换第 `mid` 个和第 `right` 个元素, 并将 `right` 指针左移
 - 若 `nums[mid]` 为1: 将指针 `mid` 右移。
- 补充一下, 当 `mid` 与 `left` 交换后, `mid` 能够后移, 因为此时 `nums[mid]` 可能为0, 后面还需要与 `left` 交换。

复杂度分析

- 时间复杂度: 由于对长度 `n` 的数组进行了一次遍历, 时间复杂度为 $O(n)$ 。

- 空间复杂度: 由于只使用了常数空间, 空间复杂度为 $O(1)$ 。

代码


python

c++

java

```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm_source=code
 */
class Solution:
    """
    @param nums: A list of integer which is 0, 1 or 2
    @return: nothing
    """
    def sortColors(self, nums):
        left = 0
        right = len(nums) - 1
        mid = left

        # 直到mid > right停止遍历
        while mid <= right:
            if nums[mid] == 0:
                nums[mid], nums[left] = nums[left], nums[mid]
                left += 1
                mid += 1
            elif nums[mid] == 2:
                nums[mid], nums[right] = nums[right], nums[mid]
                right -= 1
            else:
                mid += 1
```

 获赞 1 1 条评论

华助教

更新于 6/9/2020, 7:03:57 AM

```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm\_source=code
 */
/**
 * @param nums: A list of integer which is 0, 1 or 2
 * @return:
 */
const sortColors = function (nums) {
  if (nums === null || nums.length <= 1) {
    return;
  }
  var pl = 0;
  var pr = nums.length - 1;
  var i = 0;
  var tmp;
  while (i <= pr) {
    if (nums[i] === 0) {
      tmp = nums[i];
      nums[i] = nums[pl];
      nums[pl] = tmp;
      pl++;
      i++;
    } else if (nums[i] === 1) {
      i++;
    } else {
      tmp = nums[i];
      nums[i] = nums[pr];
      nums[pr] = tmp;
      pr--;
    }
  }
}
```

👍 获赞 1

💬 添加评论



Di

更新于 12/10/2020, 2:24:54 AM

非常简单易懂的方法, $O(n)$ 。利用多个指针分别记录各自颜色应该插入的位置。

```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ /
Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm\_source=code
 */
class Solution:
    """
    @param: nums: A list of integer which is 0, 1 or 2
    @return: nothing
    """
    def sortColors(self, nums):
        # write your code here
        i = 0
        j = 0
        for n in range(len(nums)):
            v = nums[n]
            nums[n] = 2
            if v < 2:
                nums[j] = 1
                j += 1
            if v == 0:
                nums[i] = 0
                i += 1
```

👍 获赞 4

💬 1 条评论



九章用户9NFA7E

更新于 10/14/2020, 9:39:00 PM

use classic pivot partition template twice first time to keep left all 0, e.g. 0, 0, 2, 1, 2 second time to keep left all 1 e.g. 0, 0, 1, 2, 2

```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm\_source=code
 */
class Solution:
    """
    @param nums: A list of integer which is 0, 1 or 2
    @return: nothing
    """
    def sortColors(self, nums):
        # write your code here

        ## partition 0 to left
        start = self.pivot_partition(nums, 1, 0)

        ## partition 1 to left
        self.pivot_partition(nums, 2, start)

    def pivot_partition(self, nums, pivot, start):

        l, r = start, len(nums) - 1

        while l <= r:
            # keep nums[l] < pivot staying in place, move l
            while l <= r and nums[l] < pivot:
                l += 1
            # keep nums[r] >= pivot staying in place, move r
            while l <= r and nums[r] >= pivot:
                r -= 1
            # swap num[l], num[r]
            if l <= r:
                nums[l], nums[r] = nums[r], nums[l]
                l += 1
                r -= 1

        return l
```

👍 获赞 4 💬 添加评论



九章用户BJJVP5

更新于 6/9/2020, 7:03:47 AM

简单直观暴力解法 真的没啥说明的 不知道令狐老师看到有何感想

```
/**
 * 本参考程序由九章算法用户提供。版权所有，转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作，授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括：九章算法班 2020升级版，算法强化班，算法基础班，北美算法面试高频题班，Java 高级工程师 P6+ 小班课，面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括：系统设计 System Design，面向对象设计 OOD
 * - 专题及项目类课程包括：动态规划专题班，Big Data - Spark 项目实战，Django 开发项目课
 * - 更多详情请见官方网站：http://www.jiuzhang.com/?utm_source=code
 */
public class Solution {
    /**
     * @param nums: A list of integer which is 0, 1 or 2
     * @return: nothing
     */
    public void sortColors(int[] nums) {
        // write your code here
        int red = 0;
        int blue = 0;
        int white = 0;

        for (int i = 0; i < nums.length; i++) {
            if (nums[i] == 0) {
                red++;
            }

            if (nums[i] == 1) {
                blue++;
            }

            if (nums[i] == 2) {
                white++;
            }
        }

        for (int i = 0; i < red; i++) {
            nums[i] = 0;
        }

        for (int i = red; i < red + blue; i++) {
            nums[i] = 1;
        }

        for (int i = red + blue; i < nums.length; i++) {
            nums[i] = 2;
        }
    }
}
```

👍 获赞 4

💬 6 条评论



Jeff

更新于 6/9/2020, 7:03:48 AM

哈哈哈哈哈

```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm\_source=code
 */
public class Solution {
    /**
     * @param nums: A list of integer which is 0, 1 or 2
     * @return: nothing
     */
    public void sortColors(int[] nums) {
        // write your code here
        Arrays.sort(nums);
    }
}
```

👍 获赞 3

💬 6 条评论



九章用户T12FVG

更新于 6/9/2020, 7:03:50 AM

两次 two pointer 方法 我觉得第一个同学那个 统计数量的方法非常creative
可以我评论不了 一直失败。。。


```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm_source=code
 */
public class Solution {
    /**
     * @param nums: A list of integer which is 0, 1 or 2
     * @return: nothing
     */
    public void sortColors(int[] nums) {
        // write your code here
        if( nums == null || nums.length < 2) return;
        int len = nums.length;
        int end = len - 1;
        int start = 0;

        for (int i = 0; i <= end; i++) {
            while (nums[end] == 2){
                end--;
            }
            if (nums[i] == 2 && i < end) {
                nums[i] = nums[end];
                nums[end--] = 2;
            }
        }

        for (int i = end; i >= start; i--) {
            while (nums[start] == 0) {
                start++;
            }
            if (nums[i] == 0 && i > start) {
                nums[i] = nums[start];
                nums[start++] = 0;
            }
        }
    }
}
```

👍 获赞 2

💬 添加评论



九章用户QP2MNS

更新于 6/9/2020, 7:03:49 AM

使用一次扫描的办法。设立三根指针, left, index, right。定义如下规则:

left 的左侧都是 0 (不含 left) right 的右侧都是 2 (不含 right) index 从左到右扫描每个数, 如果碰到 0 就丢给 left, 碰到 2 就丢给 right。碰到 1 就跳过不管。

```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm_source=code
 */
public class Solution {
    /**
     * @param nums: A list of integer which is 0, 1 or 2
     * @return: nothing
     */
    public void sortColors(int[] nums) {
        if (nums == null || nums.length == 0) {
            return;
        }

        int pl = 0, pr = nums.length - 1;
        int index = 0;
        while (index <= pr) {
            if (nums[index] == 0) {
                swap(nums, pl, index);
                index++;
                pl++;
            }
            else if (nums[index] == 2) {
                swap(nums, pr, index);
                pr--;
            }
            else {
                index++;
            }
        }

        private void swap(int[] nums, int a, int b) {
            int temp = nums[a];
            nums[a] = nums[b];
            nums[b] = temp;
        }
    }
}
```

👍 获赞 2 💬 添加评论



D同学

更新于 11/19/2020, 1:10:10 AM

传统的two pointer两次partition, 和模板代码类似, 不过要写两遍。但是很无脑就可以写出来

```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm_source=code
 */
public void sortColors(int[] nums) {

    int i = 0, j = nums.length-1;

    while (i <= j) {
        while (i <= j && nums[i] != 2)
            i++;
        while (i <= j && nums[j] == 2)
            j--;
        if (i <= j) {
            int temp = nums[i];
            nums[i] = nums[j];
            nums[j] = temp;
            i++;
            j--;
        }
    }

    i = 0;
    j = nums.length-1;
    while (i <= j) {
        while (i <= j && nums[i] != 1)
            i++;
        while (i <= j && nums[j] != 0)
            j--;
        if (i <= j) {
            int temp = nums[i];
            nums[i] = nums[j];
            nums[j] = temp;
            i++;
            j--;
        }
    }
}
```

👍 获赞 1 💬 添加评论



九章用户**GE5JQH**

更新于 10/5/2020, 1:53:21 AM

使用兩次循環。第一次將0跟1,2交換, 使得分成各一邊, 第二次將1,2交換, 使得1,2各一邊

```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm\_source=code
 */
class Solution:
    def sortColors(self, nums):

        if nums is None or len(nums) == 0:
            return

        left, right = 0, len(nums) - 1
        while left < right:
            while left < right and nums[left] == 0:
                left += 1
            while left < right and nums[right] >= 1:
                right -= 1

            if left < right:
                nums[left], nums[right] = nums[right], nums[left]
                left += 1
                right -= 1

        left, right = 0, len(nums) - 1
        while left < right:
            while left < right and nums[left] <= 1:
                left += 1
            while left < right and nums[right] == 2:
                right -= 1

            if left < right:
                nums[left], nums[right] = nums[right], nums[left]
                left += 1
                right -= 1
```

👍 获赞 1

💬 添加评论



九章令狐冲

更新于 11/4/2020, 1:27:26 AM

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.

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

click to show follow up.

Follow up: A rather straight forward solution is a two-pass algorithm using counting sort. First, iterate the array counting number of 0's, 1's, and 2's, then overwrite array with total number of 0's, then 1's and followed by 2's.


Could you come up with an one-pass algorithm using only constant space?

c++

java

python

```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm_source=code
 */
class Solution:
    """
    @param nums: A list of integer which is 0, 1 or 2
    @return: nothing
    """
    def sortColors(self, nums):
        left, right = 0, len(nums) - 1
        middle = 0
        while middle <= right:
            if nums[middle] == 0:
                nums[middle], nums[left] = nums[left], nums[middle]
                left += 1
                middle += 1
            elif nums[middle] == 1:
                middle += 1
            else:
                nums[middle], nums[right] = nums[right], nums[middle]
                right -= 1
```

 获赞 0 添加评论**九章令狐冲**

更新于 11/4/2020, 1:05:40 AM

做两次 Partition。先把0和非0分开, 再把1和非1分开。

python

java

```

/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm_source=code
 */
class Solution:
    """
    @param nums: A list of integer which is 0, 1 or 2
    @return: nothing
    """
    def sortColors(self, A):
        index = self.sort(A, 0, 0)
        self.sort(A, 1, index)

    def sort(self, A, flag, index):
        start, end = index, len(A) - 1
        while start <= end:
            while start <= end and A[start] == flag:
                start += 1
            while start <= end and A[end] != flag:
                end -= 1
            if start <= end:
                A[start], A[end] = A[end], A[start]
                start += 1
                end -= 1
        return start

```

👍 获赞 0 💬 添加评论

SmallTeddy

更新于 8/20/2020, 11:12:26 PM

```

/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm_source=code
 */
sortColors=function (nums) {
    return nums.sort()
}

```

👍 获赞 0 💬 添加评论

九章用户X8R5MA

更新于 6/9/2020, 7:04:27 AM

```
/**
 * 本参考程序由九章算法用户提供。版权所有，转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作，授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括：九章算法班 2020升级版，算法强化班，算法基础班，北美算法面试高频题班，Java 高级工程师 P6+ 小班课，面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括：系统设计 System Design，面向对象设计 OOD
 * - 专题及项目类课程包括：动态规划专题班，Big Data - Spark 项目实战，Django 开发项目课
 * - 更多详情请见官方网站：http://www.jiuzhang.com/?utm_source=code
 */
class Solution{
public:
    /**
     * @param nums: A list of integer which is 0, 1 or 2
     * @return: nothing
     */
    void sortColors(vector<int> &nums) {
        if (nums.empty()) {
            return;
        }
        int n = nums.size();
        // 先把{1,2}当成一类，{0}当成另一类，做一次划分
        int pos_nzero = 0;
        for (int i = 0; i < n; ++i) {
            if (nums[i] == 0) {
                std::swap(nums[i], nums[pos_nzero++]);
            }
        }

        // 此时pos_nzero指向第一个{1,2}出现的位置，再以此为起点对{1,2}做一次划分
        int pos_no1 = pos_nzero;
        for (int i = pos_nzero; i < n; ++i) {
            if (nums[i] == 1) {
                std::swap(nums[i], nums[pos_no1++]);
            }
        }
    }
};
```

👍 获赞 0

💬 添加评论



九章用户GE5JQH

更新于 6/9/2020, 7:04:27 AM

使用老師上課講的方法，類似解答的Java版奔，只進行一次循環就分類好

```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm\_source=code
 */
class Solution:
    """
    @param: nums: A list of integer which is 0, 1 or 2
    @return: nothing
    """
    def sortColors(self, nums):
        if not nums or len(nums) == 0:
            return

        i, left, right = 0, 0, len(nums) - 1
        while i <= right:
            if nums[i] == 0:
                nums[left], nums[i] = nums[i], nums[left]
                left += 1
                i += 1
            elif nums[i] == 1:
                i += 1
            else:
                nums[i], nums[right] = nums[right], nums[i]
                right -= 1
```

👍 获赞 0

💬 1 条评论



熊熊熊

更新于 6/9/2020, 7:04:27 AM

参考答案代码虽然精简, 但是有很多swap是不必要的swap, 我对此做了一点优化。


```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm_source=code
 */
public class Solution {
    /**
     * @param nums: A list of integer which is 0, 1 or 2
     * @return: nothing
     */
    public void sortColors(int[] nums) {
        // write your code here
        if (nums == null || nums.length < 2){
            return;
        }
        int i0 = 0;
        int i2 = nums.length - 1;
        int p = 0;
        while (p <= i2){
            if (nums[p] == 0){
                if (p != i0){
                    nums[p] = 1;
                    nums[i0] = 0;
                }
                p++;
                i0++;
                continue;
            }
            if (nums[p] == 1){
                p++;
                continue;
            }
            if (nums[p] == 2){
                if (nums[i2] == 2){
                    i2--;
                    continue;
                }
                if (nums[i2] == 1){
                    nums[p++] = 1;
                    nums[i2--] = 2;
                    continue;
                }
                if (nums[i2] == 0){
                    nums[p++] = 1;
                    nums[i0++] = 0;
                    nums[i2--] = 2;
                    continue;
                }
            }
        }
    }
}
```

👍 获赞 0

💬 1 条评论

**jessiew**

更新于 6/9/2020, 7:04:24 AM

two pointers基础上去掉一些不必要的swap, 加了文字演示方便理解~

/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm_source=code
 */

```

public class Solution {
    public void sortColors(int[] nums) {
        // Edge cases
        if (nums == null || nums.length == 0) return;

        // Initial state: [0(pRed),1(pWhite),0,2,1,0,2,2(pBlue)]
        int pRed = 0, pWhite = pRed + 1, pBlue = nums.length - 1;

        while (pWhite < pBlue) {
            // Move pWhite to not 1: [0(pRed),1,0(pWhite),2,1,0,2,2(pBlue)]
            while (pWhite < pBlue && nums[pWhite] == 1) pWhite++;

            /* If pWhite points to zero:
            * Move pRed to non-zero ==> [0,1(pRed),0(pWhite),2,1,0,2,2(pBlue)]
            * swap(pRed, pWhite) ==> [0,0(pRed),1(pWhite),2,1,0,2,2(pBlue)]
            */
            if (pWhite < pBlue && nums[pWhite] == 0) {
                while (pRed < pWhite && nums[pRed] == 0) pRed++;
                if (pRed < pWhite) swap(nums, pRed, pWhite);
            }

            // Make sure pWhite is always at the right side of pRed
            if (pWhite <= pRed) pWhite = pRed + 1;

            /* If pWhite points to two:
            * Move pBlue to non-two ==> [0,0(pRed),1,2(pWhite),1,0(pBlue),2,2]
            * Swap(pWhite, pBlue) ==> [0,0(pRed),1,0(pWhite),1,2(pBlue),2,2]
            */
            if (pWhite < pBlue && nums[pWhite] == 2) {
                while (pWhite < pBlue && nums[pBlue] == 2) pBlue--;
                if (pWhite < pBlue) swap(nums, pWhite, pBlue);
            }
        }

        private void swap(int[] nums, int i, int j) {
            int temp = nums[i];
            nums[i] = nums[j];
            nums[j] = temp;
        }
    }
}

```

👍 获赞 0

💬 添加评论



kurumi

更新于 6/9/2020, 7:04:22 AM

说实话真的不如rainbowSort直接做, 简单又暴力, 省的各种swap。。。。。

邀请有礼

vitation/shi



```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm_source=code
 */
class Solution {
public:
    /**
     * @param nums: A list of integer which is 0, 1 or 2
     * @return: nothing
     */
    void sortColors(vector<int> &nums) {
        // write your code here
        if (nums.empty()) {
            return;
        }
        rainbowSort(nums, 0, nums.size() - 1, 0, 2);
    }

private:
    void rainbowSort(vector<int> &nums, int start, int end, int colorFrom, int colorTo) {
        if (colorFrom == colorTo) {
            return;
        }
        if (start >= end) {
            return;
        }

        int left = start, right = end;
        int colorMid = colorFrom + (colorTo - colorFrom) / 2;

        while (left <= right) {
            while (left <= right && nums[left] <= colorMid) {
                left++;
            }
            while (left <= right && nums[right] > colorMid) {
                right--;
            }

            if (left <= right) {
                int temp = nums[left];
                nums[left] = nums[right];
                nums[right] = temp;
                left++;
                right--;
            }
        }

        rainbowSort(nums, start, right, colorFrom, colorMid);
        rainbowSort(nums, left, end, colorMid + 1, colorTo);
    }
};
```

👍 获赞 0

💬 1 条评论



九章用户57Y194

更新于 6/9/2020, 7:04:11 AM

用题干中challenge的做法, 数一下有多少个0, 1, 2。然后把整个array override。代码也很好懂。

```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ /
Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm\_source=code
 */
def sortColors(self, nums):
    # write your code here
    if not nums:
        return None
    count_0, count_1, count_2 = 0, 0, 0
    for i in range(len(nums)):
        if nums[i] == 0:
            count_0 += 1
        elif nums[i] == 1:
            count_1 += 1
        elif nums[i] == 2:
            count_2 += 1
    for a in range(count_0):
        nums[a] = 0
    for b in range(count_0, count_0+count_1):
        nums[b] = 1
    for c in range(count_0+count_1, len(nums)):
        nums[c] = 2
```

👍 获赞 0

💬 添加评论



九章用户F76MHX

更新于 6/9/2020, 7:04:06 AM

增加一个小附件, 让每次p至少从p1出发, 这样可以避免数组前面好几个0出现从而使得而p停留在原地的状况

```

/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm_source=code
 */
class Solution:
    """
    @param nums: A list of integer which is 0, 1 or 2
    @return: nothing
    """
    def sortColors(self, nums):
        p1, p2 = 0, len(nums) - 1
        p = 0
        while p <= p2:
            if nums[p] == 0:
                p1 += 1
                p = max(p, p1)
                continue
            if nums[p] == 2:
                p2 -= 1
                continue
            if nums[p] == 1:
                nums[p], nums[p1] = nums[p1], nums[p]
            elif nums[p] == 2:
                nums[p], nums[p2] = nums[p2], nums[p]
            else:
                p += 1

```

👍 获赞 0

💬 添加评论



九章用户FSHNM9

更新于 6/9/2020, 7:04:05 AM

用循环不变式的方式对三个指针的做法进行了一些推导。假设初始状态*l*, *m*, *r*指向0, *r*指向*n*-1, 我们可以认为在最后一排完序的情况下 1) 有且只有0, *l* 的元素都是0。因为一开始本身不能保证是0, 但我可以保证排完之后*l*以前的都是0 2) 有且只有(*r*, *end* () 的元素都是2。同理因为一开始*r*本身并不是2, 但我可以保证排完之后*r*以后的都是2 3) 有且只有[*l*, *m*) 的元素都是1。因为排完的时候最后一个0后面*l*指向的元素得是个1。*m*一开始不是1, 但是可以保证排完之后*m*以前到*l*元素的都是1 4) 有且只有*m*, *r* () 的元素就是需要swap的工作区间

这样来看比较明显*m*的值就很重要了, *m*必须动的合理准确才能完成排序。继续分情况讨论, 在工作区间*m*, *r* () 中 1) *numm* () == 0。我们已经知道[0, *l*) 必须是0, 也就是说*numl* () 现在不是0, 那*m*就和*l*换一下。换完之后*l*上的元素变成了0, 那么*l*需要往后移一位。重点在这里。我们又知道[*l*, *m*) 保证是1, 说明刚才从*l*换到*m*上的元素肯定是一个1。这时*m*也需要往后移一位才能满足[*l*, *m*) 都是1的要求 2) *numm* () == 1。 *m*直接往后移保证[*l*, *m*) 都是1就完了。 3) *numm* () == 2。我们知道(*r*, *end*] 必须是2, 也就是说*numr* () 现在不是2, 那*m*就和*r*换一下。换完之后*r*上的元素变成了2, 那么*r*需要往前移一位保证(*r*, *end*] 都是2。这时从*r*换到*m*的元素可能是0或1, 并没有任何条件的限制, 所以*m*不能动

从工作区间是*m*, *r* () 可以知道在*m* == *r* 的情况下也是需要判断并且做相应操作的所以循环的condition是while(*m* <= *r*)

```
/**
 * 本参考程序由九章算法用户提供。版权所有，转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作，授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括：九章算法班 2020升级版，算法强化班，算法基础班，北美算法面试高频题班，Java 高级工程师 P6+ 小班课，面试软技能指导 - BQ /
Resume / Project 2020版
 * - Design类课程包括：系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括：动态规划专题班，Big Data - Spark 项目实战，Django 开发项目课
 * - 更多详情请见官方网站：http://www.jiuzhang.com/?utm_source=code
 */
class Solution {
public:
    void sortColors(vector<int> &nums) {
        // write your code here
        int l = 0, r = nums.size()-1, m = 0;
        while (m <= r) {
            if (nums[m] > 1) {
                swap(nums[m], nums[r]);
                r--;
            } else if (nums[m] < 0) {
                swap(nums[m], nums[l]);
                l++;
                m++;
            } else {
                m++;
            }
        }
    }
};
```

 获赞 0 添加评论**Jet**

更新于 6/9/2020, 7:04:04 AM

Sort Color.Sort Color.Sort Color.Sort Color.

```
/**
 * 本参考程序由九章算法用户提供。版权所有，转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作，授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括：九章算法班 2020升级版，算法强化班，算法基础班，北美算法面试高频题班，Java 高级工程师 P6+ 小班课，面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括：系统设计 System Design，面向对象设计 OOD
 * - 专题及项目类课程包括：动态规划专题班，Big Data - Spark 项目实战，Django 开发项目课
 * - 更多详情请见官方网站：http://www.jiuzhang.com/?utm_source=code
 */
class Solution {
public:
    /**
     * @param nums: A list of integer which is 0, 1 or 2
     * @return: nothing
     */
    void sortColors(vector<int> &nums) {
        // write your code here
        if(nums.size()==0||nums.size()==1){
            return;
        }
        int left=0;
        int right=nums.size()-1;
        int index=0;
        while(index<=right){
            if(nums[index]==0){
                // int temp=nums[index];
                nums[index]=nums[left];
                nums[left]=0;
                left++;
                index++;
                continue;
            }
            if(nums[index]==1){
                index++;
                continue;
            }
            //nums[index]==2
            nums[index]=nums[right];
            nums[right]=2;
            right--;
        }

        return;
    }
};
```

👍 获赞 0 💬 添加评论



我要AC

更新于 6/9/2020, 7:04:00 AM

彩虹排序。

time: $O(n\log(3))$ space: $O(1)$

```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ /
Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm\_source=code
 */
class Solution:
    """
    @param nums: A list of integer which is 0, 1 or 2
    @return: nothing
    """
    def sortColors(self, nums):
        # write your code here

        """
        solution: rainbowsort
        time: O(nlogk) == O(nlog3)
        space: O(1)
        """
        def partition(colors, start, end, color_start, color_end):
            if start >= end or color_start==color_end:
                return

            color_pivot = (color_start+color_end)//2
            left, right = start, end
            while left <= right:
                # print('{} {}'.format(left, right))
                while left <= right and colors[left]<=color_pivot:
                    left+=1
                while left <= right and colors[right]>color_pivot:
                    right-=1
                if left <= right:
                    colors[left], colors[right] = colors[right], colors[left]
                    left+=1
                    right-=1

            partition(colors, start, right, color_start, color_pivot)
            partition(colors, left, end, color_pivot+1, color_end)

        partition(nums, 0, len(nums)-1, 0, 2)
```

👍 获赞 0

💬 添加评论

进阶课程

视频+互动	直播+互动	直播+互动	互动课
<div>九章算法班 2021 版</div> <div>8周时间精通 57 个核心高频考点，9招击破 FLAG、BATJ 算法面试。22....</div>	<div>系统架构设计 System Design 2021 版</div> <div>成为百万架构师必上。30 课时带你快速掌握18大系统架构设计知识点与面...</div>	<div>九章算法面试高频题冲刺班</div> <div>每期更新 15% 题目，考前押题，一举拿下FLAG & BATJ Offer</div>	<div>面向对象设计 OOD</div> <div>应届生及亚马逊面试必考，IT求职必备基础</div>