

LintCode领扣题解 (/problem) / 数组划分 · Partition Array

数组划分 · Partition Array

中文

LintCode 版权所有 (/problem/?tags=lintcode-copyright)

两根指针 (/problem/?tags=two-pointers)

数组 (/problem/?tags=array)

sort (/problem/?tags=sort)

描述

给出一个整数数组 *nums* 和一个整数 *k*。划分数组（即移动数组 *nums* 中的元素），使得：

- 所有小于*k*的元素移到左边
- 所有大于等于*k*的元素移到右边

返回数组划分的位置，即数组中第一个位置 *i*，满足 *nums[i]* 大于等于 *k*。

❗ 你应该真正的划分数组 **nums**，而不仅仅是计算比 **k** 小的整数数，如果数组 **nums** 中的所有元素都比 **k** 小，则返回 *nums.length*。

样例

例1:

输入：
[], 9
输出：
0

例2:

输入：
[3, 2, 2, 1], 2
输出: 1
解释：
真实的数组为 [1, 2, 2, 3]。所以返回 1

挑战

使用 $O(n)$ 的时间复杂度在数组上进行划分。

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

收起题目描述 ^

语言类型

ALL (28)

java (12)

python (8)

cpp (7)

javascript (1)

上传题解



九章-小原

更新于 11/29/2020, 8:22:38 PM

算法：快速选择算法

通过头尾指针跳过小于 *k* 的前缀和大于等于 *k* 的后缀，可以找到与第一个大于等于 *k* 的值和最后一个小于 *k* 的值。进行交换后可达到划分数组的目的，直到找到两个指针相遇为止。

伪代码如下：

- 令 $left = 0$, $right = length - 1$ 。
- 当 $nums[left] < k$ 时, $left$ 指针向右移动。
- 当 $nums[right] \geq k$ 时, $right$ 指针向左移动。
- 如果 $left \leq right$, 交换两个值。
- 如果 $left > right$, 返回 $left$ 作为最终结果, 否则返回第二步。

复杂度分析

设数组长度为 n

时间复杂度 $O(n)$

- 最多扫描一遍数组, 时间复杂度为 $O(n)$

空间复杂度 $O(1)$

- 只需要选择用的游标变量的额外空间, 所以空间复杂度为 $O(1)$

源代码

java

c++

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: The integer array you should partition
    @param k: An integer
    @return: The index after partition
    """
    def partitionArray(self, nums, k):
        left, right = 0, len(nums) - 1

        while left <= right:
            while left <= right and nums[left] < k:
                left += 1
            while left <= right and nums[right] >= k:
                right -= 1
            if left <= right:
                nums[left], nums[right] = nums[right], nums[left]
                left += 1
                right -= 1

        return left
```

👍 获赞 1

💬 添加评论

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

**华助教**

更新于 6/9/2020, 7:04:26 AM

快速选择算法

```
/**
 * 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: The integer array you should partition
 * @param k: An integer
 * @return: The index after partition
 */
const partitionArray = function (nums, k) {
    if (nums === null || nums.length === 0) {
        return 0;
    }
    var left = 0, right = nums.length - 1, temp;
    while (left <= right) {
        while (left <= right && nums[left] < k) {
            left++;
        }
        while (left <= right && nums[right] >= k) {
            right--;
        }
        if (left <= right) {
            temp = nums[left];
            nums[left] = nums[right];
            nums[right] = temp;
            left++;
            right--;
        }
    }
    return left;
}
```

👍 获赞 0

💬 添加评论

**九章用户GFRHAY**

更新于 7/21/2020, 2:15:13 PM

空间复杂度: $O(n)$ 非常简单的算法: 设置一个offset值, 每次将 $<k$ 的值与offset位置上的值交换位置, 然后offset++。因为不需要担心顺序, 所以可以无论exchange的数是否小于k, 最后所有左边的数都会小于k

```
/**
 * 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: The integer array you should partition
     * @param k: An integer
     * @return: The index after partition
     */
    public int partitionArray(int[] nums, int k) {
        // write your code here
        if (nums == null || nums.length == 0) {
            return 0;
        }

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

        return offset;
    }
}
```

👍 获赞 9 💬 添加评论



JasonHao

更新于 10/2/2020, 4:31:20 PM

既然只有一个分界线，那就两个指针，一个指向第一个大于等于k的也可以说是即将可能要被swap的，一个循环数组只要超过k就和前面指针swap。

```
/**
 * 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 partitionArray(self, nums, k):

    left = 0
    for i in range(len(nums)):
        if nums[i] < k:
            nums[left], nums[i] = nums[i], nums[left]
            left += 1

    return left
```

👍 获赞 7 💬 2 条评论

**九章令狐冲**

更新于 12/26/2020, 5:29:32 PM

快速选择算法

python

java

c++

```
/**
 * 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: The integer array you should partition
    @param k: As description
    @return: The index after partition
    """
    def partitionArray(self, nums, k):
        left, right = 0, len(nums) - 1
        while left <= right:
            while left <= right and nums[left] < k:
                left += 1
            while left <= right and nums[right] >= k:
                right -= 1
            if left <= right:
                nums[left], nums[right] = nums[right], nums[left]
                left += 1
                right -= 1
        return left
```

👍 获赞 2

💬 添加评论

**BrickMover**

更新于 12/17/2020, 7:44:14 PM

31. Partition Array ()

```

/**
 * 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
 */
"""

简洁的根据pivot对数组进行分区的算法:
若n=len(nums), 则每一次while循环结束后nums[0]到nums[i]是确定<k的, nums[i+1]到nums[j]是确定>=k的, nums[j+1]到nums[n-1]是等待与k进行比较的。
当nums[i] < k, 只增加j, 否则先交换nums[i]和nums[j]然后分别增加i和j。
极端情况下(所有数都<k或所有数都>=k) 代码依然适用。
时间复杂度: O(n)。空间复杂度: O(1)。
"""

def partitionArray(self, nums, k):
    if len(nums)==0:
        return 0
    i, j, n = 0, 0, len(nums)
    while j < n:
        if nums[j] < k:
            nums[i], nums[j] = nums[j], nums[i]
            i += 1
        j += 1
    return i

```

👍 获赞 2 💬 添加评论



Serena

更新于 6/9/2020, 7:03:55 AM


left指针记录比k小的数, right指针记录比k大的数, 最后返回k+1就是最终的位置

```

/**
 * 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: The integer array you should partition
    @param k: An integer
    @return: The index after partition
    """
    def partitionArray(self, nums, k):
        # write your code here
        if nums==[]:
            return 0
        left,right=0,len(nums)-1
        for i in range(0,len(nums)):
            if nums[i]< k:
                left+=1
            elif nums[i]>=k:
                right-=1

        return right+1

```

 获赞 1 1 条评论**Mengjie**

更新于 11/19/2020, 8:29:37 AM

本来想用双指针 奈何想了想改了改还是暴力出来了

```
/**
 * 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: The integer array you should partition
     * @param k: An integer
     * @return: The index after partition
     */
    public int partitionArray(int[] nums, int k) {
        // write your code here
        if(nums == null || nums.length == 0){
            return 0;
        }

        int start = 0;
        int end = nums.length - 1;
        Arrays.sort(nums);

        if(k <= nums[start]){
            return 0;
        }
        if(k > nums[end]){
            return nums.length;
        }

        for(int i = 0; i < nums.length; ++i){
            if(nums[i] != k){
                continue;
            }
            return i;
        }
        return 0;
    }
}
```

 获赞 0 添加评论**EricC**

更新于 10/23/2020, 11:00:29 PM

小幅度修改自九章partition模板, 仅1处改动。O(n), 201ms. 具体看题目中的注释

```

/**
 * 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: The integer array you should partition
     * @param k: An integer
     * @return: The index after partition
     */
    public int partitionArray(int[] nums, int k) {
        // write your code here
        if (nums == null || nums.length == 0)
            return 0;

        /**
         * Similar to No.143.
         * Note that there is only 1 chance to move all numbers, therefore,
         * 1. do not left++ after swap, give the existence of duplicates, imagine a duplicated number is swapped,
         *    then left should stay here pointing to the duplicated number.
         * 2. right-- after swap, this is safe, and would avoid infinite loop that keep swapping duplicates.
         * 3. right will always be 1 less than the k, even -1, so either
         *    return right + 1
         *    or left, since left stays at original position after swap.
         */

        int left = 0, right = nums.length - 1;
        while (left <= right) {
            while (left <= right && nums[left] < k) {
                left++;
            }
            while (left <= right && nums[right] > k) {
                right--;
            }
            if (left <= right) {
                int temp = nums[right];
                nums[right] = nums[left];
                nums[left] = temp;
                // left++;
                right--;
            }
        }
        return left;
    }
}

```

👍 获赞 0


💬 2 条评论

[\(/\) 课程 \(/course/\)](#)
[旗舰课 \(/premium-course/\)](#)
[1对1私教 \(/1on1/\)](#)
[免费课](#)
[seminar](#)
[APP](#)
[邀请有礼](#)
[刷题题解](#)
[成功案](#)
[消息](#)
[更多...](#)
[我的课程 \(/accounts/profile/\)](#)

 九章用户H32DMG

更新于 9/7/2020, 12:05:08 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
 */
public class Solution {
    public int partitionArray(int[] nums, int k) {
        if(nums.length==0){
            return 0;
        }
        int i=0,j=0;
        for(int num:nums){
            if(num<k){
                int temp = nums[i];
                nums[i] = nums[j];
                nums[j] = temp;
                i++;
            }
            j++;
        }
        return i;
    }
}
```

 获赞 0 添加评论

vitation/sh:



小陈同学

更新于 8/13/2020, 3:17:11 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
 */
public class Solution {
    /**
     * @param nums: The integer array you should partition
     * @param k: An integer
     * @return: The index after partition
     */
    public int partitionArray(int[] nums, int k) {
        // write your code here
        if (nums == null || nums.length == 0) {
            return 0;
        }
        int index = 0;
        int kPos = 0;
        while (index < nums.length) {
            if (nums[index] < k) {
                swap(nums, index, kPos);
                kPos++;
            }
            index++;
        }

        return kPos;
    }

    private void swap(int nums[], int pos1, int pos2) {
        int temp = nums[pos1];
        nums[pos1] = nums[pos2];
        nums[pos2] = temp;
    }
}
```

👍 获赞 0

💬 添加评论



Yinglao

更新于 6/9/2020, 7:04:26 AM

分治法：处理两段已经partition好的数列，从右半部分的pr = k - 1和左半部分的pl = k 分别往后和往前，逐个替换，直到pl大于middleIndex 或者pr 小于等于 middleIndex Basecase: endIndex = startIndex, 如果该元素>=k, return startIndex, 否则return startIndex + 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: The integer array you should partition
    @param k: An integer
    @return: The index after partition
    """
    def partitionArray(self, nums, k):
        # write your code here
        if not nums or len(nums) == 0:
            return 0
        return self.helper(nums, k, 0, len(nums) - 1)

    def helper(self, nums, k, si, ei):
        if si == ei:
            return si if nums[si] >= k else si + 1
        mi = si + (ei - si) // 2
        pl = self.helper(nums, k, si, mi)
        pr = self.helper(nums, k, mi + 1, ei) - 1
        while pl <= mi and pr > mi:
            self.swap(nums, pl, pr)
            pl += 1
            pr -= 1
        return pr + 1 if pl > mi else pl

```

👍 获赞 0

💬 添加评论

加载更多题解

进阶课程

视频+互动

直播+互动

直播+互动

互动课

九章算法班 2021 版

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....

系统架构设计 System Design 2021 版

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...

九章算法面试高频题冲刺班

每期更新 15% 题目, 考前押题, 一举拿下FLAG & BATJ Offer

面向对象设计 OOD

应届生及亚马逊面试必考, IT求职必备基础

