

LintCode领扣题解 (/problem) / 中位数 · Median

## 中位数 · Median

中文

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

快速排序 (/problem/?tags=quick-sort)

数组 (/problem/?tags=array)

### 描述

给定一个未排序的整数数组，找到其中位数。

中位数是排序后数组的中间值，如果数组的个数是偶数个，则返回排序后数组的第 $N/2$ 个数。

① 数组大小不超过10000

### 样例

样例 1:

输入: [4, 5, 1, 2, 3]  
输出: 3  
解释:  
经过排序，得到数组 [1, 2, 3, 4, 5]，中间数字为3

样例 2:

输入: [7, 9, 4, 5]  
输出: 5  
解释:  
经过排序，得到数组 [4, 5, 7, 9]，第二个 ( $4/2$ ) 数字为5

### 挑战

时间复杂度为 $O(n)$

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

收起题目描述 ^

语言类型

ALL (15)

java (7)

python (5)

c++ (2)

javascript (1)

上传题解



令狐冲

更新于 6/9/2020, 7:03:58 AM

考点:

- 区间第k大元素的查找

题解: 使用优先队列维护大小为k的有序集合，优先队列大小小于k时直接压入，等于k时与首个元素判断大小关系，小于时直接压入，并将首个元素弹出。最后输出优先队列的头部元素即可。优先队列:

- 优先队列中，元素被赋予优先级。当访问元素时，具有最高优先级的元素最先删除。优先队列具有最高级先出的行为特征。通常采用堆数据结构来实现。
- 默认情况下，优先队列按照非递增顺序排列。

```
/**
 * 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 integers.
     * @return: An integer denotes the middle number of the array.
     */
    int median(vector<int> &nums) {
        // write your code here
        int k = (nums.size() + 1) / 2;
        priority_queue<int> que;
        int len = nums.size();
        for(int i = 0; i < len; i++) {
            if(que.size() == k) {
                if(nums[i] < que.top()) {
                    que.pop();
                    que.push(nums[i]);
                }
            } else {
                que.push(nums[i]);
            }
        }
        return que.top();
    }
};
```

👍 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



九章算法助教团队

更新于 6/9/2020, 7:03:53 AM

考点:

- 区间第k大元素的查找

题解: 基于快排的分治思想, 递归实现, 从而复杂度接近 $O(n)$ 。

```

/**
 * 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 integers
    @return: An integer denotes the middle number of the array
    """
    def median(self, nums):
        # write your code here
        if not nums or len(nums) == 0:
            return

        return self.sortQuick(nums, 0, len(nums) - 1, (len(nums) + 1) // 2)

    def sortQuick(self, nums, start, end, k):
        if start == end:
            return nums[start]

        left, right = start, end
        pivot = nums[left + (right - left) // 2]
        while left <= right:
            while left <= right and nums[left] < pivot:
                left += 1
            while left <= right and nums[right] > pivot:
                right -= 1
            if left <= right:
                nums[left], nums[right] = nums[right], nums[left]
                left += 1
                right -= 1

        if start + k - 1 <= right:
            return self.sortQuick(nums, start, right, k)
        if start + k - 1 >= left:
            return self.sortQuick(nums, left, end, k - (left - start))

        return nums[right + 1]

```

👍 获赞 1

💬 添加评论



令狐冲

更新于 6/9/2020, 7:04:31 AM

考点:

- 区间第k大元素的查找

题解: 利用快速排序的思想指导, 不断递归将一个区间分为左右两个小区间, 当枢轴左区间大小为所求中位数位置时, 寻找到返回值。经过一次划分之后, 枢轴pivot将原序列划分为两个部分: S和T pivot包含在子序列S中 (), 会出现下列三种情况:

```
/**
 * 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
 */
·子序列S中有K个数, 此时pivot位置即为第K大的数, 返回

·子序列S中的数字个数小于K, 假设个数为L, 则需要子序列T中继续递归划分出前(K-L)个数

·子序列S中的数字个数大于K, 则需要子序列S中继续递归划分出前K个数
```

```
/**
 * 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 integers.
     * @return: An integer denotes the middle number of the array.
     */
    public int median(int[] nums) {
        return sub(nums, 0, nums.length - 1, (nums.length + 1)/2);
    }
    private int sub(int[] nums, int start, int end, int size) {
        int mid = (start + end) / 2;
        int pivot = nums[mid];
        int i = start - 1, j = end + 1;
        for (int k = start; k < j; k++) {
            if (nums[k] < pivot) {
                i++;
                int tmp = nums[i];
                nums[i] = nums[k];
                nums[k] = tmp;
            } else if (nums[k] > pivot) {
                j--;
                int tmp = nums[j];
                nums[j] = nums[k];
                nums[k] = tmp;
                k--;
            }
        }
        if (i - start + 1 >= size) {
            return sub(nums, start, i, size);
        } else if (j - start >= size) {
            return nums[j-1];
        } else {
            return sub(nums, j, end, size - (j - start));
        }
    }
}
```

👍 获赞 0

💬 1 条评论

**令狐冲**

更新于 6/9/2020, 7:04:31 AM

- python选择使用内置的排序算法, 在有序数组中寻找中位数直接输出

```
/**
 * 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 integers.
    @return: An integer denotes the middle number of the array.
    """
    def median(self, nums):
        # write your code here
        nums.sort()
        return nums[(len(nums)-1)/2]
```

获赞 0

7 条评论

**华助教**

更新于 6/9/2020, 7:04:26 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 integers
 * @return: An integer denotes the middle number of the array
 */
const median = function (nums) {
  sub = function(nums, start, end, size) {
    var mid = Math.floor((start + end) / 2);
    var pivot = nums[mid];
    var i = start - 1, j = end + 1, k;
    var tmp;
    for (k = start; k < j; k++) {
      if (nums[k] < pivot) {
        i++;
        tmp = nums[i];
        nums[i] = nums[k];
        nums[k] = tmp;
      } else if (nums[k] > pivot) {
        j--;
        tmp = nums[j];
        nums[j] = nums[k];
        nums[k] = tmp;
        k--;
      }
    }
    if (i - start + 1 >= size) {
      return sub(nums, start, i, size);
    } else if (j - start >= size) {
      return nums[j-1];
    } else {
      return sub(nums, j, end, size - (j - start));
    }
  }
  return sub(nums, 0, nums.length - 1, Math.floor((nums.length + 1)/2));
}
```

👍 获赞 0    💬 添加评论



九章用户4GDNUB

更新于 6/9/2020, 7:03:46 AM

O(n) 时间复杂度的，基于快速排序 partition 模板的 median 实现。

```
/**
 * 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 {
    /**
     * @author WildCat
     * @param nums: A list of integers
     * @return: An integer denotes the middle number of the array
     */
    public int median(int[] nums) {
        // write your code here

        // Please note: the `k` must be (nums.length + 1) / 2 - 1 in order to satisfy the requirement
        return partition(nums, 0, nums.length - 1, (nums.length + 1) / 2 - 1);
    }

    // Same to find kth smallest element in an array
    // Modified for the partition function of quicksort
    private int partition(int[] nums, int left, int right, int k) {
        if (left == right) {
            return nums[k];
        }

        int start = left;
        int end = right;
        int pivot = nums[(end - start) / 2 + start];
        while (start <= end) {
            while (nums[start] < pivot && start <= end) {
                start++;
            }
            while (nums[end] > pivot && start <= end) {
                end--;
            }
            if (start <= end) {
                int temp = nums[start];
                nums[start] = nums[end];
                nums[end] = temp;
                start++;
                end--;
            }
        }

        if (k <= end) {
            return partition(nums, left, end, k);
        } else if (k >= start) {
            return partition(nums, start, right, k);
        }
        return nums[k];
    }
}
```

👍 获赞 5

💬 2 条评论



九章用户QP2MNS

更新于 6/9/2020, 7:03:49 AM

使用类似于第k大元素, 使用quickSelect实现求中位数。

```

/**
 * 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 integers
     * @return: An integer denotes the middle number of the array
     */
    public int median(int[] nums) {
        // write your code here
        if (nums == null || nums.length == 0) {
            return -1;
        }
        int n = nums.length;
        int k = n % 2 == 0 ? n / 2 : n / 2 + 1;
        return quickSelect(nums, 0, n - 1, k - 1);
    }
    private int quickSelect(int[] A, int start, int end, int k) {
        if (start == end) {
            return A[start];
        }
        int left = start, right = end;
        while (left <= right) {
            while (left <= right && A[left] < pivot) {
                left++;
            }
            while (left <= right && A[right] > pivot) {
                right--;
            }
            if (left <= right) {
                int tmp = A[left];
                A[left] = A[right];
                A[right] = tmp;
                left++;
                right--;
            }
        }
        if (k >= start && k <= right) {
            return quickSelect(A, start, right, k);
        }
        if (k >= left && k <= end) {
            return quickSelect(A, left, end, k);
        }
        return A[right + 1];
    }
}

```

👍 获赞 2

💬 添加评论



Owen

更新于 6/9/2020, 7:03:56 AM

用了quickSelect, 算法班的那个模板去写的。跟kth-largest-element一样。



```
/**
 * 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 - R0 / 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 integers
     * @return: An integer denotes the middle number of the array
     */
    public int median(int[] nums) {
        // write your code here
        if(nums == null || nums.length == 0){
            return -1;
        }

        return quickSelect(nums, 0, nums.length - 1, (nums.length - 1) / 2);
    }

    private int quickSelect(int[] nums, int start, int end, int medianIndex){
        if(start == end){
            return nums[start];
        }

        int i = start, j = end;
        int pivot = nums[(start + end) / 2];
        while(i <= j){
            while(i <= j && nums[i] < pivot){
                i++;
            }
            while(i <= j && nums[j] > pivot){
                j--;
            }
            if(i <= j){
                int temp = nums[i];
                nums[i] = nums[j];
                nums[j] = temp;
                i++;
                j--;
            }
        }

        if(start + medianIndex <= j){
            return quickSelect(nums, start, j, medianIndex);
        }
        if(start + medianIndex >= i){
            return quickSelect(nums, i, end, medianIndex - (i - start));
        }
        return nums[j + 1];
    }
}
```

👍 获赞 1

💬 4 条评论



Tin

更新于 6/9/2020, 7:03:54 AM

真正的统计意义上的O(N)实现。Random Select。

另, 在递归里变换求第几个数, 即中位数的指针是自找麻烦, 左右指针都没变 (还是绝对值), 中位数指针变来变去 (变成相对值) 也让函数的参数失去一致性。

```
/**
 * 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
 */
import random
class Solution:

    def median(self, nums) -> int:

        random.seed()
        return self.quick_select((len(nums)-1)//2, nums, 0, len(nums)-1)

    def quick_select(self, mid, nums, left, right):

        if left == right: return nums[left]

        pivot = nums[random.randint(left, right)]
        lo, hi = left, right

        while lo <= hi:

            while lo <= hi and nums[lo] < pivot:
                lo += 1
            while lo <= hi and nums[hi] > pivot:
                hi -= 1

            if lo <= hi:
                nums[lo], nums[hi] = nums[hi], nums[lo]
                lo += 1
                hi -= 1

        if mid <= hi:
            return self.quick_select(mid, nums, left, hi)

        if mid >= lo:
            return self.quick_select(mid, nums, lo, right)

        return nums[mid]
```

👍 获赞 1

💬 添加评论



九章用户SW4YZL

更新于 6/9/2020, 7:03:52 AM

完全按照partition的模板来写, 把前k大换成的k设置成  $k = \text{median}$

精选答案中index的设置改来改去, 这个完全无脑模板

```

/**
 * 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 median(self, nums):
        if not nums:
            return

        return self.partition(nums, 0, len(nums) - 1, (len(nums) - 1) // 2)

    def partition(self, nums, start, end, k):
        if start == end:
            return nums[start]

        left, right = start, end
        pivot = nums[(start + end) // 2]

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

        if k <= right:
            return self.partition(nums, start, right, k)
        if k >= left:
            return self.partition(nums, left, end, k)

        return nums[k]

```

👍 获赞 1      💬 添加评论



leetcode

更新于 6/9/2020, 7:04:26 AM

minHeap and quickSelect <http://www.lintcode.com/zh-cn/problem/median/#> (<http://www.lintcode.com/zh-cn/problem/median/#>)

```

/**
 * 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 : A list of integers
     * @return: An integer denotes the middle number of the array
     */
    public int median(int[] nums) {
        if (nums == null || nums.length == 0) {

```

```

        return 0;
    }
    int l = 0, r = nums.length - 1, mid = (nums.length + 1)/2;
    while (true) {
        int nth = partition(nums, l, r);
        if (nth + 1 == mid) {
            return nums[nth];
        } else if (nth + 1 > mid) {
            r = nth - 1;
        } else {
            l = nth + 1;
        }
    }
}

public int partition(int[] nums, int start, int end) {
    int l = start, r = end, pivot = nums[start];
    while (l <= r) {
        while (l <= r && nums[l] <= pivot) {
            l++;
        }
        while (l <= r && nums[r] >= pivot) {
            r--;
        }
        if (l <= r) {
            swap(nums, l++, r--);
        }
    }
    swap(nums, start, r);
    return r;
}

public void swap(int[] nums, int l, int r) {
    int temp = nums[r];
    nums[r] = nums[l];
    nums[l] = temp;
    return;
}

// public int median(int[] nums) {
//     // write your code here
//     if (nums == null || nums.length == 0) {
//         return 0;
//     }

//     // nums.length    mid(minHeap)
//     // 4                3
//     // 5                3

//     int mid = 0;

//     if (nums.length % 2 == 0) {
//         mid = (nums.length)/ 2 + 1;
//     } else {
//         mid = (nums.length + 1)/ 2;
//     }

//     PriorityQueue<Integer> minHeap = new PriorityQueue<>();
//     for (int n : nums) {
//         minHeap.offer(n);
//         if (minHeap.size() > mid) {
//             minHeap.poll();
//         }
//     }
//     return minHeap.poll();
// }

```

}

👍 获赞 0

💬 添加评论



leetcode

更新于 6/9/2020, 7:04:26 AM

<http://www.lintcode.com/zh-cn/problem/median/#> (<http://www.lintcode.com/zh-cn/problem/median/#>) for-loop

```
/**
 * 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 : A list of integers
     * @return: An integer denotes the middle number of the array
     */
    public int median(int[] nums) {
        if (nums == null || nums.length == 0) {
            return 0;
        }
        int l = 0, r = nums.length - 1, mid = (nums.length + 1)/2;
        while (true) {
            int nth = partition(nums, l, r);
            if (nth + 1 == mid) {
                return nums[nth];
            } else if (nth + 1 > mid) {
                r = nth - 1;
            } else {
                l = nth + 1;
            }
        }
    }

    public int partition(int[] nums, int start, int end) {
        int pivot = nums[end], index = start;
        for (int i = start; i < end; i++) {
            if (nums[i] <= pivot) {
                swap(nums, i, index++);
            }
        }
        swap(nums, end, index);
        return index;
    }

    public void swap(int[] nums, int l, int r) {
        int temp = nums[r];
        nums[r] = nums[l];
        nums[l] = temp;
        return;
    }
}
```

👍 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

[首页 \(/?skip\\_redirect=true\)](#) | [联系我们 \(mailto:info@jiuzhang.com\)](mailto:info@jiuzhang.com) | [加入我们 \(/joinus\)](#)

Copyright © 2013-2020 九章算法 浙ICP备19045946号-1  
(<http://www.miibeian.gov.cn/>)

商务合作: [fukesu@jiuzhang.com \(mailto:fukesu@jiuzhang.com\)](mailto:fukesu@jiuzhang.com)

 (<http://weibo.com/ninechapter>)  知  
(<https://www.zhihu.com/people/crackinterview/>)

(/)