

LintCode领扣题解 (/problem) / Quick Sort · Quick Sort

Quick Sort · Quick Sort

中文

描述

收起题目描述 ^

语言类型

ALL (7)

java (3)

python (3)

cpp (1)

上传题解

九章用户0417BW

更新于 6/9/2020, 7:03:46 AM

Java实现的快速排序算法 .



invitation/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
 */
import java.util.Random;

public class Solution {
    /**
     * @param A: an integer array
     * @return:
     */
    public Random rand;
    public void sortIntegers2(int[] A) {
        rand = new Random();
        // write your code here
        quickSort(A, 0, A.length - 1);
    }

    public void quickSort(int[] A, int start, int end) {
        if (start >= end) {
            return;
        }

        int index = rand.nextInt(end - start + 1) + start;
        int pivot = A[index];
        int left = start;
        int right = end;

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

            if (left <= right) {
                int temp = A[left];
                A[left] = A[right];
                A[right] = temp;

                left++;
                right--;
            }
        }
        // A[start... right]
        quickSort(A, start, right);
        // A[left ... end]
        quickSort(A, left, end);
    }
}
```

👍 获赞 6

💬 1 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

令狐冲

更新于 10/28/2020, 10:36:16 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
 */
代码
```

Python实现的快速排序 .

```
/**
 * 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 {int[]} A an integer array
    # @return nothing
    def sortIntegers2(self, A):
        # Write your code here
        self.quickSort(A, 0, len(A) - 1)

    def quickSort(self, A, start, end):
        if start >= end:
            return

        left, right = start, end
        # key point 1: pivot is the value, not the index
        pivot = A[(start + end) // 2]

        # key point 2: every time you compare left & right, it should be
        # left <= right not left < right
        while left <= right:
            while left <= right and A[left] < pivot:
                left += 1
            while left <= right and A[right] > pivot:
                right -= 1
            if left <= right:
                A[left], A[right] = A[right], A[left]
                left += 1
                right -= 1

        self.quickSort(A, start, right)
        self.quickSort(A, left, end)
```

👍 获赞 4

💬 1 条评论

令狐冲

更新于 9/6/2020, 7:35:32 PM

快速排序的经典实现方法 可以去领扣上提交 Sort Integers 这个题 pivot 取中点还是随机取这个关系不大。一般取中点就好了

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
 */
public class Solution {
    /**
     * @param A: an integer array
     * @return:
     */
    public void sortIntegers2(int[] A) {
        quickSort(A, 0, A.length - 1);
    }

    public void quickSort(int[] A, int start, int end) {
        if (start >= end) {
            return;
        }

        int pivot = A[start + (end - start) / 2];
        int left = start;
        int right = end;

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

            if (left <= right) {
                int temp = A[left];
                A[left] = A[right];
                A[right] = temp;

                left++;
                right--;
            }
        }
        // sort A[start... right]
        quickSort(A, start, right);
        // sort A[left ... end]
        quickSort(A, left, end);
    }
}

```

👍 获赞 1

💬 添加评论



ryan

更新于 6/9/2020, 7:04:19 AM

基于以数组端点为pivot的quick sort。两种方法分别为pivot = Aleft () 和 pivot = Aright ()。

```

/**
 * 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](http://www.jiuzhang.com/?utm_source=code)

\*/

```
public class Solution {
    /**
     * @param A: an integer array
     * @return: nothing
     */
    public void sortIntegers2(int[] A) {
        quickSort(A, 0, A.length - 1);
    }

    private void swap(int[]A, int i, int j) {
        int tmp = A[i];
        A[i] = A[j];
        A[j] = tmp;
    }

    // method 1 : right point pivot
    private void quickSort(int[] A, int l, int r) {
        if (l >= r) {
            return;
        }

        int loc = l - 1;
        int pivot = A[r];

        for (int i = l; i < r; i++) {
            if (A[i] < pivot) {
                loc++;
                swap(A, i, loc);
            }
        }
        loc++;
        swap(A, r, loc);

        quickSort(A, l, loc - 1);
        quickSort(A, loc + 1, r);
    }

    // method 2 : left point pivot
    private void quickSort(int[] A, int l, int r) {
        if (l >= r) {
            return;
        }

        int loc = r + 1;
        int pivot = A[l];

        for (int i = r; i > l; i--) {
            if (A[i] >= pivot) {
                loc--;
                swap(A, i, loc);
            }
        }
        loc--;
        swap(A, l, loc);

        quickSort(A, l, loc - 1);
        quickSort(A, loc + 1, r);
    }
}
```

👍 获赞 0

💬 添加评论

**Tommy**

更新于 6/9/2020, 7:04:12 AM

Quick sort in C++. Any comment is welcome.

```
/**
 * 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
 */
vector<int> sortArray(vector<int>& nums) {
    if (nums.size() <= 1) return nums;
    qSort(nums, 0, nums.size() - 1);
    return nums;
}

void qSort(vector<int>& nums, int left, int right) {
    if (left >= right) return;
    int k = partition(nums, left, right);
    qSort(nums, left, k);
    qSort(nums, k + 1, right);
}

int partition(vector<int>& nums, int left, int right) {
    int i = left;
    int j = right;
    int mid = (i + j) >> 1;
    int pivot = nums[mid];

    while (i <= j) {
        while (i <= j && nums[i] < pivot) i++;
        while (i <= j && nums[j] > pivot) j--;
        if (i <= j) {
            swap(nums[i++], nums[j--]);
        }
    }
    return j <= left? left : j;
}
```

👍 获赞 0

💬 添加评论

**Yuchen**

更新于 6/9/2020, 7:04:11 AM

在原数组上操作快排 help function只涉及数组要操作部分的左右端点两个参数

```
/**
 * 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
 */
class Solution(object):
    def quickSort(self, array):
        """
        input: int[] array
        return: int[]
        """
        # write your solution here

    if len(array) <= 1:
        return array

    def helper(left, right): # a helper function only include the left and right index as input

        if left >= right:
            return array[left:right+1]

        pivot = array[right]

        j = left
        for i in range(left, right):
            if array[i] < pivot:
                array[j], array[i] = array[i], array[j]
                j += 1

        array[left:j] = helper(left, j-1)
        array[j+1:right+1] = helper(j, right-1)
        array[j] = pivot
        # change the numbers at the left of j and right of j and finally change j

        return array[left:right+1]

    left = 0
    right = len(array) - 1

    return helper(left, right)
```

👍 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 &amp; 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>