

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

Merge Sort · Merge Sort

中文

描述

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

收起题目描述 ^


语言类型

ALL (4)

java (2)

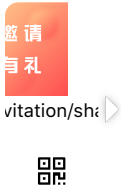
python (2)

上传题解



令狐冲

更新于 10/30/2020, 12:07:03 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 A an integer array
     * @return void
     */
    public void sortIntegers2(int[] A) {
        // use a shared temp array, the extra memory is O(n) at least
        int[] temp = new int[A.length];
        mergeSort(A, 0, A.length - 1, temp);
    }

    private void mergeSort(int[] A, int start, int end, int[] temp) {
        if (start >= end) {
            return;
        }

        int left = start, right = end;
        int mid = (start + end) / 2;

        mergeSort(A, start, mid, temp);
        mergeSort(A, mid+1, end, temp);
        merge(A, start, mid, end, temp);
    }

    private void merge(int[] A, int start, int mid, int end, int[] temp) {
        int left = start;
        int right = mid+1;
        int index = start;

        // merge two sorted subarrays in A to temp array
        while (left <= mid && right <= end) {
            if (A[left] < A[right]) {
                temp[index++] = A[left++];
            } else {
                temp[index++] = A[right++];
            }
        }
        while (left <= mid) {
            temp[index++] = A[left++];
        }
        while (right <= end) {
            temp[index++] = A[right++];
        }

        // copy temp back to A
        for (index = start; index <= end; index++) {
            A[index] = temp[index];
        }
    }
}
```

👍 获赞 4

💬 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

Boolean

更新于 6/9/2020, 7:03:48 AM

mergeSort 非递归算法

```
/**
 * 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
 */
public class Solution {

    public static void mergeSort(int[] a){
        int[] TR = new int[a.length]; //用于存放归并结果

        int k=1; //起始, 子序列长度为1
        while(k<a.length){
            mergePass(a, TR, k, a.length); //将原先无序的数据两两归并入TR
            k = 2*k; //子序列长度加倍
            mergePass(TR, a, k, a.length); //将TR中已经两两归并的有序序列再归并回数组a
            k = 2*k; //子序列长度加倍
        }
    }

    public static void mergePass(int[] SR, int [] TR, int s, int len){

        int i=0;
        while (i < len-2*s+1) { //8
            merge(SR, TR, i, i+s-1, i+2*s-1); //两两归并
            i=i+2*s;
        }

        //处理最后的尾数
        //i=8
        if(i< len-s+1){ //9
            merge(SR, TR, i, i+s-1, len-1); //归并最后两个序列
        } else {
            for (int j = i; j < len; j++) { //若最后只剩下单个子序列
                TR[j] = SR[j];
            }
        }
    }

    public static void merge(int[] SR, int[] TR, int i, int m, int n){
        int j, k, l;

        //i(0~4) j(5~8)
        for(j=m+1, k=i; i<=m && j<=n; k++){

            if(SR[i]<SR[j]){
                TR[k] = SR[i++];
            } else {
                TR[k] = SR[j++];
            }
        }
    }
}
```

```
        if(i<=m){
            for (l = 0; l <= m-i ; l++) {
                TR[k+l] = SR[i+l];
            }
        }

        if(j<=n){
            for (l = 0; l <= n-j; l++) {
                TR[k+l] = SR[j+l];
            }
        }
    }

    public static void main(String[] args) {
        int[] a = {2,3,5,4,1,6,9,8,7,10,20,45,32,28,44,31,55,43,23,21,23,21,33,21};
        mergeSort(a);
        for(int i=0;i<a.length;i++)
            System.out.print(a[i]+" ");
    }
}
```

👍 获赞 3

💬 1 条评论



九章用户0417BW

更新于 11/19/2020, 12:56:41 PM

Python实现的归并排序 .

```
/**
 * 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
        temp = [0 for _ in range(len(A))]
        self.merge_sort(0, len(A) - 1, A, temp)

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

        mid = (start + end) // 2
        self.merge_sort(start, mid, A, temp)
        self.merge_sort(mid + 1, end, A, temp)
        self.merge(start, mid, end, A, temp)

    def merge(self, start, mid, end, A, temp):
        left, right = start, mid + 1
        index = start
        while left <= mid and right <= end:
            if A[left] < A[right]:
                temp[index] = A[left]
                left += 1
            else:
                temp[index] = A[right]
                right += 1

            index += 1

        while left <= mid:
            temp[index] = A[left]
            left += 1
            index += 1

        while right <= end:
            temp[index] = A[right]
            right += 1
            index += 1

        for index in range(start, end + 1):
            A[index] = temp[index]
```

👍 获赞 1

💬 4 条评论

**Larry**

更新于 12/17/2020, 7:54:38 PM

<https://www.jianshu.com/p/3ad5373465fd> (<https://www.jianshu.com/p/3ad5373465fd>) 这么写起来可能简洁一点

```
/**
 * 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 mergesort(self, seq):
    """归并排序"""
    if len(seq) <= 1:
        return seq
    mid = len(seq) // 2 # 将列表分成更小的两个列表
    # 分别对左右两个列表进行处理, 分别返回两个排序好的列表
    left = self.mergesort(seq[:mid])
    right = self.mergesort(seq[mid:])
    # 对排序好的两个列表合并, 产生一个新的排序好的列表
    return self.merge(left, right)

def merge(self, left, right):
    """合并两个已排序好的列表, 产生一个新的已排序好的列表"""
    result = [] # 新的已排序好的列表
    i = 0 # 下标
    j = 0
    # 对两个列表中的元素 两两对比。
    # 将最小的元素, 放到result中, 并对当前列表下标加1
    while i < len(left) and j < len(right):
        if left[i] <= right[j]:
            result.append(left[i])
            i += 1
        else:
            result.append(right[j])
            j += 1
    result += left[i:]
    result += right[j:]
    return result
```

👍 获赞 3

💬 添加评论

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

