

dreamcatcher-cx

why is more important than what.

博客园

首页

新随笔

联系

订阅

管理

图解排序算法(五)之快速排序——三数取中法

快速排序由C. A. R. Hoare在1962年提出。它的基本思想是：通过一趟排序将要排序的数据分割成独立的两部分，其中一部分的所有数据都比另外一部分的所有数据都要小，然后再按此方法对这两部分数据分别进行快速排序，整个排序过程可以递归进行，以此达到整个数据变成有序序列

基本步骤

三数取中

在快排的过程中，每一次我们要取一个元素作为枢纽值，以这个数字来将序列划分为两部分。在此我们采用三数取中法，也就是取左端、中间、右端三个数，然后进行排序，将中间数作为枢纽值。

公告

昵称： dreamcatcher-cx
园龄： 4年7个月
粉丝： 1066
关注： 33
[+加关注](#)

< 2021年4月 >						
日	一	二	三	四	五	六
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1
2	3	4	5	6	7	8

积分与排名

积分 - 63297

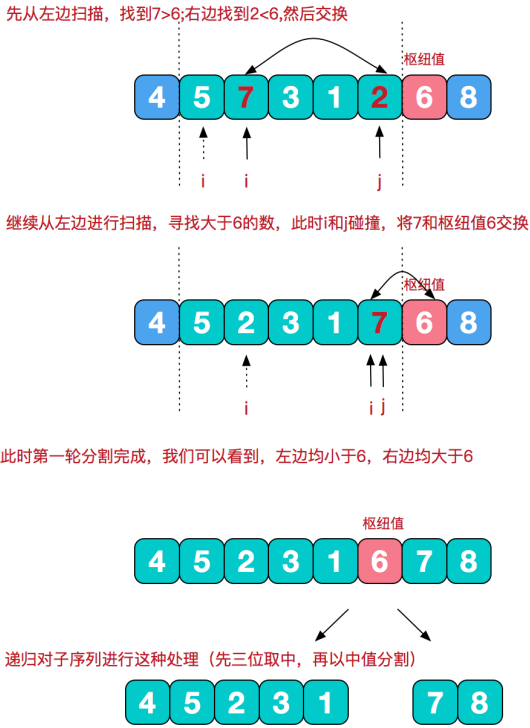
排名 - 17684

随笔分类 (20)



根据枢纽值进行分割

双向扫描，从左边找大于枢纽值的数，从右边找小于枢纽值的数，然后交换之。
由于我们的枢纽值在右边，所以要先从左边开始扫描



java集合框架(1)

Oracle(4)

并发编程(8)

数据结构(2)

算法(5)

随笔档案 (20)

2017年7月(2)

2017年6月(1)

2017年5月(2)

2017年4月(1)

2017年3月(1)

2017年2月(1)

2017年1月(1)

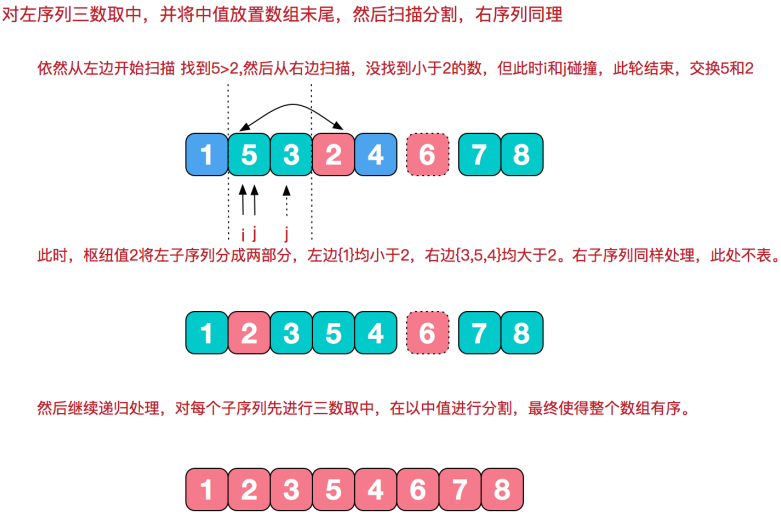
2016年12月(3)

2016年11月(4)

2016年10月(2)

2016年9月(2)

最新评论



代码实现

```
package sortdemo;

import java.util.Arrays;

/**
 * Created by chengxiao on 2016/12/14.
 * 快速排序
 */
public class QuickSort {
    public static void main(String[] args) {
        int[] arr = {9, 8, 7, 6, 5, 4, 3, 2, 1, 0};
        quickSort(arr, 0, arr.length - 1);
        System.out.println("排序结果: " + Arrays.toString(arr));
    }

    /**
     * @param arr
     * @param left 左指针
     * @param right 右指针
     */
    public static void quickSort(int[] arr, int left, int right) {
        if (left < right) {
            //获取枢纽值，并将其放在当前待处理序列末尾
            dealPivot(arr, left, right);

            //枢纽值被放在序列末尾
            int pivot = right - 1;
```

1. Re:图解排序算法(四)之归并排序

算法小白，其实有一个比较疑惑的问题，相比插入/冒泡的排序算法，使用分治思想的排序算法在开始排序之前还要经过一个拆分数组的过程，总共的时间加起来，真的会比冒泡快吗，还是说随着数组长度变化，长度跟时间的曲...

--去骨鸡腿排

2. Re:图解排序算法(四)之归并排序

初始化 t=0有问题吧

t=left

--java渣渣

3. Re:图解排序算法(二)之希尔排序

@Boblim 博主理解是对的，有写到实际实现时不需要严格按照分组进行，这样能减少一层你写的for(int k=0;k<div;++k)分组循环，你可以模拟运行再体会下。你的代码虽然是严格按照算法理解...

--stagelovepig

4. Re:ConcurrentHashMap实现原理及源码分析

很棒

--收纸箱易拉罐

5. Re:图解排序算法(二)之希尔排序

不停的跳组,分组插入排序,相同的分组之间就是排序两个有序数组,感觉和归并差不多啊(狗头).

--吕思豪

```
//左指针
int i = left;
//右指针
int j = right - 1;
while (true) {
    while (arr[++i] < arr[pivot]) {
    }
    while (j > left && arr[--j] > arr[pivot]) {
    }
    if (i < j) {
        swap(arr, i, j);
    } else {
        break;
    }
}
if (i < right) {
    swap(arr, i, right - 1);
}
quickSort(arr, left, i - 1);
quickSort(arr, i + 1, right);
}

/**
 * 处理枢纽值
 *
 * @param arr
 * @param left
 * @param right
 */
public static void dealPivot(int[] arr, int left, int
right) {
    int mid = (left + right) / 2;
    if (arr[left] > arr[mid]) {
        swap(arr, left, mid);
    }
    if (arr[left] > arr[right]) {
        swap(arr, left, right);
    }
    if (arr[right] < arr[mid]) {
        swap(arr, right, mid);
    }
    swap(arr, right - 1, mid);
}
```

阅读排行榜

- 1. 图解排序算法(三)之堆排序(563527)
- 2. 图解排序算法(四)之归并排序(355483)
- 3. HashMap实现原理及源码分析(332762)
- 4. 图解排序算法(一)之3种简单排序(选择, 冒泡, 直接插入)(240501)
- 5. 图解排序算法(二)之希尔排序(222641)

评论排行榜

- 1. HashMap实现原理及源码分析(70)
- 2. 图解排序算法(三)之堆排序(67)
- 3. 图解排序算法(四)之归并排序(47)
- 4. 图解排序算法(二)之希尔排序(27)
- 5. 图解排序算法(一)之3种简单排序(选择, 冒泡, 直接插入)(18)

推荐排行榜

- 1. HashMap实现原理及源码分析(167)
- 2. 图解排序算法(三)之堆排序(166)
- 3. 图解排序算法(四)之归并排序(132)
- 4. ConcurrentHashMap实现原理及源码分析(49)

```
/**
 * 交换元素通用处理
 *
 * @param arr
 * @param a
 * @param b
 */
private static void swap(int[] arr, int a, int b) {
    int temp = arr[a];
    arr[a] = arr[b];
    arr[b] = temp;
}
```



排序结果

排序结果: [1, 2, 3, 4, 5, 6, 7, 8]

总结

快速排序是一种交换类的排序，它同样是分治法的经典体现。在一趟排序中将待排序的序列分割成两组，其中一部分记录的关键字均小于另一部分。然后分别对这两组继续进行排序，以使整个序列有序。在分割的过程中，枢纽值的选择至关重要，本文采取了三位取中法，可以很大程度上避免分组“一边倒”的情况。快速排序平均时间复杂度也为 $O(n\log n)$ 级。

作者: dreamcatcher-cx

出处: <<http://www.cnblogs.com/chengxiao/>>

本文版权归作者和博客园共有，欢迎转载，但未经作者同意必须保留此段声明，且在页面明显位置给出原文链接。

分类: 算法

好文要顶

关注我

收藏该文



dreamcatcher-cx

关注 - 33

粉丝 - 1066

17

0

+加关注

« 上一篇: 图解排序算法(四)之归并排序

» 下一篇: 数据结构(二)之二叉树

posted @ 2017-01-08 18:12 dreamcatcher-cx 阅读(63256) 评论(11)

[编辑](#) [收藏](#)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

登录后才能查看或发表评论, 立即 [登录](#) 或者 [逛逛](#) [博客园首页](#)

【推荐】阿里云云小站限量代金券, 新老用户同享, 上云优惠聚集地

【推荐】大型组态、工控、仿真、CAD\GIS 50万行VC++源码免费下载!

【推荐】#悄悄变强大# 五一假期提升指南, 你若学习, 机会自来

【推荐】限时秒杀! 国云大数据魔镜, 企业级云分析平台

园子动态:

- 致园友们的一封检讨书: 都是我们的错
- 数据库实例 CPU 100% 引发全站故障
- 发起一个开源项目: 博客引擎 fluss

最新新闻:

- 法官力挺亚马逊 拒绝驳回干预AWS采购100亿美元国防合同的投诉
 - Facebook向广告商详细阐述苹果ATT对他们的影响
 - 特斯拉全新MPV渲染图曝光 造型汽车史上绝无仅有
 - 大部分欧美游戏开发者不认为Steam应获得30%的收入
 - 首战告捷! 中国空间站出征太空 3年后或全球唯一
- » [更多新闻...](#)

Copyright © 2021 dreamcatcher-cx

Powered by .NET 5.0 on Kubernetes