

昵称: 紫红的泪

园龄: 6年11个月

粉丝: 141

关注: 1

+加关注

龙渊阁

博客园 首页 新随笔 联系 订阅 管理

随笔 - 182 文章 - 2 评论 - 119

3017814284

18101920

726273

714121

24252627

28293031123

45678910

搜索

找找看

谷歌搜索

常用链接

我的随笔

我的评论

我的参与

最新评论

我的标签

随笔分类(264)

C(20)

C#(4)

C++(65)

Python(21)

Qt(22)

汉化(2)

其他(22)

启动盘(1)

日志(8)

软件(4)

算法练习(59)

游戏开发(3)

转载精华(33)

随笔档案(178)

2014年4月 (2)

2014年3月 (2)

2014年1月 (2)

2013年9月 (6)

2013年7月 (1)

2013年6月 (2)

2013年5月 (6)

2013年2月 (2)

2013年1月 (5)

2012年12月 (3)

2012年11月 (11)

2012年10月 (22)

2012年9月 (28)

2012年8月 (27)

2012年6月 (1)

2012年3月 (1)

2011年11月 (1)

2011年8月 (1)

2011年4月 (4)

2011年3月 (6)

2011年2月 (1)

2010年12月 (1)

2010年11月 (1)

2010年10月 (4)

2010年8月 (4)

2010年7月 (3)

2010年6月 (1)

2010年5月 (6)

2010年4月 (13)

2010年3月 (3)

2010年2月 (2)

稳定排序和不稳定排序

这几天笔试了好几次了，连续碰到一个关于常见排序算法稳定性判别的问题，往往还是多选，对于我以及和我一样拿不准的同学可不是一个能轻易下结论的题目，当然如果你笔试之前已经记住了数据结构书上哪些是稳定的，哪些不是稳定的，做起来应该可以轻松搞定。本文是针对老是记不住这个或者想真正明白到底为什么是稳定或者不稳定的人准备的。

Ai还是要在Aj位置前。

其次，说一下稳定性的好处。排序算法如果是稳定的，那么从一个键上排序，然后再从另一个键上排序，第一个键排序的结果可以为第二个键排序所用。基数排序就是这样，先按低位排序，逐次按高位排序，低位相同的元素其顺序再高位也相同时是不会改变的。另外，如果排序算法稳定，对基于比较的排序算法而言，元素交换的次数可能会少一些（个人感觉，没有证实）。

回到主题，现在分析一下常见的排序算法的稳定性，每个都给出简单的理由。

(1)冒泡排序

冒泡排序就是把小的元素往前调或者把大的元素往后调。比较是相邻的两个元素比较，交换也发生在这两个元素之间。所以，如果两个元素相等，我想你是不会再无聊地把他们俩交换一下的；如果两个相等的元素没有相邻，那么即使通过前面的两两交换把两个相邻起来，这时候也不会交换，所以相同元素的前后顺序并没有改变，所以冒泡排序是一种稳定排序算法。

(2)选择排序

选择排序是给每个位置选择当前元素最小的，比如给第一个位置选择最小的，在剩余元素里面给第二个元素选择第二小的，依次类推，直到第n - 1个元素，第n个元素不用选择了，因为只剩下它一个最大的元素了。那么，在一趟选择，如果当前元素比一个元素小，而该小的元素又出现在一个和当前元素相等的元素后面，那么交换后稳定性就被破坏了。比较拗口，举个例子，序列5 8 5 2 9，我们知道第一遍选择第1个元素5会和2交换，那么原序列中2个5的相对前后顺序就被破坏了，所以选择排序不是一个稳定的排序算法。

(3)插入排序

插入排序是在一个已经有序的小序列的基础上，一次插入一个元素。当然，刚开始这个有序的小序列只有1个元素，就是第一个元素。比较是从有序序列的末尾开始，也就是想要插入的元素和已经有序的最大者开始比起，如果比它大则直接插入在其后面，否则一直往前找直到找到它该插入的位置。如果碰见一个和插入元素相等的，那么插入元素把想插入的元素放在相等元素的后面。所以，相等元素的前后顺序没有改变，从原无序序列出去的顺序就是排好序后的顺序，所以插入排序是稳定的。

(4)快速排序

快速排序有两个方向，左边的i下标一直往右走，当a[i] <= a[center_index]，其中center_index是中枢元素的数组下标，一般取为数组第0个元素。而右边的j下标一直往左走，当a[j] > a[center_index]。如果i和j都走不动了，i <= j，交换a[i]和a[j]，重复上面的过程，直到i > j。交换a[j]和a[center_index]，完成一趟快速排序。在中枢元素和a[j]交换的时候，很有可能把前面的元素的稳定性打乱，比如序列为5 3 3 4 3 8 9 10 11，现在在中枢元素5和3（第5个元素，下标从1开始计）交换就会把元素3的稳定性打乱，所以快速排序是一个不稳定的排序算法，不稳定发生在中枢元素和a[j] 交换的时刻。

(5)归并排序

归并排序是把序列递归地分成短序列，递归出口是短序列只有1个元素（认为直接有序）或者2个序列（1次比较和交换），然后把各个有序的段序列合并成一个有序的长序列，不断合并直到原序列全部排好序。可以发现，在1个或2个元素时，1个元素不会交换，2个元素如果大小相等也没有人故意交换，这不会破坏稳定性。那么，在短的有序序列合并的过程中，稳定是否受到破坏？没有，合并过程中我们可以保证如果两个当前元素相等时，我们把处在前面的序列的元素保存在结果序列的前面，这样就保证了稳定性。所以，归并排序也是稳定的排序算法。

(6)基数排序

基数排序是按照低位先排序，然后收集；再按照高位排序，然后再收集；依次类推，直到最高位。有时候有些属性是有优先级顺序的，先按低优先级排序，再按高优先级排序，最后的次序就是高优先级高的在前，高优先级相同的低优先级高的在前。基数排序基于分别排序，分别收集，所以其是稳定的排序算法。

(7)希尔排序(shell)

希尔排序是按照不同步长对元素进行插入排序，当刚开始元素很无序的时候，步长最大，所以插入排序的元素个数很少，速度很快；当元素基本有序了，步长很小，插入排序对于有序的序列效率很高。所以，希尔排序的时间复杂度会比O(n^2)好一些。由于多次插入排序，我们知道一次插入排序是稳定的，不会改变相同元素的相对顺序，但在不同的插入排序过程中，相同的元素可能在各自的插入排序中移动，最后其稳定性就会被打乱，所以shell排序是不稳定的。

(8)堆排序

我们知道堆的结构是节点i的孩子为2 * i和2 * i + 1节点，大顶堆要求父节点大于等于其2个子节点，小顶堆要求父节点小于等于其2个子节点。在一个长为n 的序列，堆排序的过程是从第n / 2开始和其子节点共3个值选择最大（大顶堆）或者最小（小顶堆），这3个元素之间的选择当然不会破坏稳定性。但当n / 2 - 1， n / 2 - 2， ... 1这些个父节点选择元素时，就会破坏稳定性。有可能第n / 2个父节点交换把后面一个元素交换过去了，而第n / 2 - 1个父节点把后面一个相同的元素没有交换，那么这2个相同的元素之间的稳定性就被破坏了。所以，堆排序不是稳定的排序算法。

综上，得出结论：选择排序、快速排序、希尔排序、堆排序不是稳定的排序算法，而冒泡排序、插入排序、归并排序和基数排序是稳定的排序算法

http://www.cnblogs.com/codingmylife/archive/2012/10/21/2732980.html

1/3

<div>偏数学的问题，可以编程解 TopCoder 网站做的太复杂了，题很好 结构之法 算法之道 面试题很多很杂，质量参差不齐 水木社区->算法版 贴的问题针对性比较强 未名空间->待字闺中版 海外名企面经 一亩三分地->算法版 有少量大公司新面试题 阅微堂 有很多奇怪的问题</div> <div>朋友博客 BlueCode hopeztm 我的摄影小站 需要Smugmug优惠码的站内私信~ 雪狼的博客 知识天地</div> <div>游戏GUI CEGUI 传说比较靠谱的开源GUI，有些商业游戏用，比如火炬之光 Scaleform 国际顶尖大作基本都是用的这个，有孤岛危机、无主之地、星际争霸2...</div> <div>游戏引擎 Cocos2d-x 不得不说这个跨各移动平台的开源C++引擎是中国人的骄傲~ KlayGE 不断更新的并且拥有最新渲染技术的开源引擎，微软亚洲研究院的大神搞的 Ogre3D 这个不多说了 Unity3D 收费的跨各种平台，但是个人感觉这玩意儿搞大型项目容易悲剧，性能调优余地不大。</div> <div>有用的链接 Check iO 用Python打游戏 David Beazley的个人主页 Python大神，他的Tutorials和Talks不错~ Eric Lippert's Blog Fabulous Adventures In Coding MAS 参与开发1年多的东西，离开MSRA很久了，打开网站满是回忆..... 酷壳 科普新闻</div> <div>积分与排名 积分 - 142409 排名 - 1461</div> <div>最新评论 1. Re:Python3.x和Python2.x的区别 好文必须顶！</div>	<div>评论列表</div> <div><div>#1楼 2014-03-26 16:02 yuanlisail</div><div>好！</div><div>支持(0) 反对(0)</div></div> <div><div>#2楼 2014-03-26 16:02 yuanlisail</div><div>转载了！</div><div>支持(0) 反对(0)</div></div> <div><div>#3楼 2015-08-29 11:15 放作夥</div><div>归并算法的合并过程中我们可以保证如果两个当前元素相等时，我们把处在前面的序列的元素保存在结果序列的前面，这样就保证了稳定性。这句话怎么理解</div><div>支持(0) 反对(0)</div></div> <div><div>#4楼 2015-10-05 17:59 William's Tech Space</div><div>学习了！</div><div>支持(0) 反对(0)</div></div> <div><div>#5楼 2016-03-09 15:16 RainyBear</div><div>楼主如果附上各种排序代更好了~~</div><div>支持(0) 反对(0)</div></div> <div><div>#6楼 2016-07-06 00:44 UnGeek</div><div>nice</div><div>支持(0) 反对(0)</div></div> <div><div>#7楼 2016-10-15 16:22 捕神</div><div>@ 放作夥 就是归并算法在合并时，等值元素不会做交换，保持了等值元素原有先后顺序。</div><div>支持(0) 反对(0)</div></div> <div><div>刷新评论 刷新页面 返回顶部</div><div>注册用户登录后才能发表评论，请 登录 或 注册，访问网站首页。</div><div>【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库</div><div><div><div>jQuery MiniUI</div><div>快速开发WEB软件，支持 Java、.Net</div></div><div>广告</div></div><div>最新IT新闻: · GitLab事故之技术详叙：抢救后恢复在线，已确定下一步计划 · 阿里巴巴成立澳大利亚、新西兰总部 · 壕出天际！马云为啥花55亿赞助奥运会？</div></div>
---	--

啄木鸟儿

2. Re:Python3.x和Python2.x的区别

不错

--linspace

3. Re:稳定排序和不稳定排序

@放作夥就是归并算法在合并时，等值元素不会做交换，保持了等值元素原有先后顺序。...

--捕神

4. Re:编译64位boost

him.exe toolset=msvc_10.0.architectu

龙渊阁

博客园 首页 新随笔 联系 订阅 管理

随笔 - 182 文章 - 2 评论 - 119

complete ==with+system.....

--洗心

5. Re:编译64位boost

需要 多加一个参数：architecture=x86

才能生成64位dll

--洗心

阅读排行榜

1. Python3.x和Python2.x的区别(139390)

2. 稳定排序和不稳定排序(48095)

3. 幻方常规解法汇总(41158)

4. C语言清空输入缓冲区的N种方法对比(36361)

5. Python: Enum枚举的实现(25844)

评论排行榜

1. C++成员函数指针错误用法警示(25)

2. Python3.x和Python2.x的区别(17)

3. 由Qmake.exe/QtCreator.exe启动速度慢挖进去(9)

4. Qt中实现单例模式（SingleTon）(8)

5. 稳定排序和不稳定排序(7)

推荐排行榜

1. Python3.x和Python2.x的区别(43)

2. 稳定排序和不稳定排序(23)

3. 由Qmake.exe/QtCreator.exe启动速度慢挖进去(4)

4. Python: Enum枚举的实现(4)

5. 编译64位boost(3)

· 最完美可口可乐上市：喝了不胖还能减肥

· 微软详解.NET语言战略：C#、VB、F#都不落

» 更多新闻...

H3 BPM

自开发 零实施的BPM

免费下载

最新知识库文章:

· 「代码家」的学习过程和学习经验分享

· 写给未来的程序媛

· 技术的正宗与野路子

» 更多知识库文章...