dreamcatcher-cx

why is more important than what.

京系 JIII **早**各回 自吸 新道軍

图解排序算法(二)之希尔排序

希尔排序是希尔(Donald Shell)于1959年提出的一种排序 算法。希尔排序也是一种插入排序,它是简单插入排序经过改进之 后的一个更高效的版本,也称为缩小增量排序,同时该算法是冲破 O(n²) 的第一批算法之一。本文会以图解的方式详细介绍希尔排 序的基本思想及其代码实现。

思不思思

希尔排序是把记录按下标的一定增量分组,对每组使用直 接插入排序算法排序;随着增量逐渐减少,每组包含的关键词 越来越多,当增量减至1时,整个文件恰被分成一组,算法便终 止。

简单插入排序很循规蹈矩,不管数组分布是怎么样的,依然一 步一步的对元素进行比较,移动,插入,比如[5,4,3,2,1,0]这种倒 序序列,数组末端的0要回到首位置很是费劲,比较和移动元素均 需n-1次。而希尔排序在数组中采用跳跃式分组的策略,通过某个 增量将数组元素划分为若干组,然后分组进行插入排序,随后逐步 缩小增量,继续按组进行插入排序操作,直至增量为1。希尔排序 通过这种策略使得整个数组在初始阶段达到从宏观上看基本有序, 小的基本在前,大的基本在后。然后缩小增量,到增量为1时,其 实多数情况下只需微调即可,不会涉及过多的数据移动。

我们来看下希尔排序的基本步骤,在此我们选择增量 gap=length/2, 缩小增量继续以gap = gap/2的方式,这种增量

公告

昵称: dreamcatcher-cx

园龄: 4年7个月 粉丝: 1066 关注: 33 +加关注

<		20)21年4	月		>
日	_	=	Ξ	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 - 63297

积分与排名

排名 - 17684

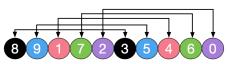
随笔分类 (20)

选择我们可以用一个序列来表示,{n/2,(n/2)/2...1}, 称为**增量序 列**。希尔排序的增量序列的选择与证明是个数学难题,我们选择的这个增量序列是比较常用的,也是希尔建议的增量,称为希尔增量,但其实这个增量序列不是最优的。此处我们做示例使用希尔增量。

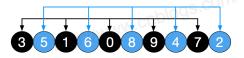
原始数组 以下数据元素颜色相同为一组



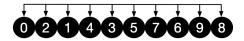
初始增量 gap=length/2=5, 意味着整个数组被分为5组, [8,3][9,5][1,4][7,6][2,0]



对这5组分别进行直接插入排序,结果如下,可以看到,像3,5,6这些小元素都被调到前面了,然后缩小增量 gap=5/2=2,数组被分为2组 [3,1,0,9,7] [5,6,8,4,2]



对以上2组再分别进行直接插入排序,结果如下,可以看到,此时整个数组的有序程度更进一步啦。 再缩小增量gap=2/2=1,此时,整个数组为1组[0,2,1,4,3,5,7,6,9,8],如下

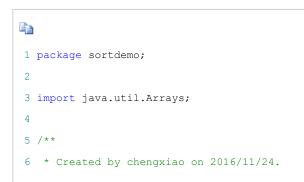


经过上面的"宏观调控",整个数组的有序化程度成果喜人。 此时,仅仅需要对以上数列简单微调,无需大量移动操作即可完成整个数组的排序。



代码实现

在希尔排序的理解时,我们倾向于对于每一个分组,逐组进行处理,但在代码实现中,我们可以不用这么按部就班地处理完一组再调转回来处理下一组(这样还得加个for循环去处理分组)比如 [5,4,3,2,1,0],首次增量设gap=length/2=3,则为3组[5,2] [4,1] [3,0],实现时不用循环按组处理,我们可以从第gap个元素开始,逐个跨组处理。同时,在插入数据时,可以采用元素交换法寻找最终位置,也可以采用数组元素移动法寻觅。希尔排序的代码比较简单,如下:



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

```
7 */
8 public class ShellSort {
      public static void main(String []args) {
10
           int []arr =\{1, 4, 2, 7, 9, 8, 3, 6\};
11
           sort(arr);
12
           System.out.println(Arrays.toString(arr));
13
          int []arr1 ={1,4,2,7,9,8,3,6};
           sort1(arr1);
14
15
           System.out.println(Arrays.toString(arr1));
16
      }
17
18
19
       * 希尔排序 针对有序序列在插入时采用交换法
20
        * @param arr
21
        * /
22
      public static void sort(int []arr){
           //增量gap, 并逐步缩小增量
23
          for(int gap=arr.length/2;gap>0;gap/=2){
24
              //从第gap个元素,逐个对其所在组进行直接插入排序操作
25
26
              for(int i=gap;i<arr.length;i++){</pre>
                  int j = i;
27
                  while(j-gap>=0 && arr[j]<arr[j-gap]){</pre>
2.8
29
                      //插入排序采用交换法
30
                      swap(arr,j,j-gap);
31
                      j-=gap;
32
33
34
35
37
        * 希尔排序 针对有序序列在插入时采用移动法。
38
39
        * @param arr
40
      public static void sort1(int []arr){
41
           //增量gap, 并逐步缩小增量
42
43
           for(int gap=arr.length/2;gap>0;gap/=2){
               //从第gap个元素,逐个对其所在组进行直接插入排序操作
44
               for(int i=gap;i<arr.length;i++) {</pre>
45
46
                   int j = i;
                   int temp = arr[j];
47
                   if(arr[j] < arr[j-gap]) {</pre>
48
                       while(j-gap>=0 && temp<arr[j-gap]){</pre>
49
                           //移动法
50
51
                           arr[j] = arr[j-gap];
```

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

```
52
                            j-=gap;
                        arr[j] = temp;
55
56
               }
           }
57
58
59
60
        * 交换数组元素
61
        * @param arr
62
        * @param a
63
        * @param b
64
       public static void swap(int []arr,int a,int b) {
65
           arr[a] = arr[a] + arr[b];
           arr[b] = arr[a] - arr[b];
           arr[a] = arr[a]-arr[b];
69
70 }
```

包括

本文介绍了希尔排序的基本思想及其代码实现,希尔排序中对于增量序列的选择十分重要,直接影响到希尔排序的性能。我们上面选择的增量序列{n/2,(n/2)/2...1}(希尔增量),其最坏时间复杂度依然为O(n²),一些经过优化的增量序列如Hibbard经过复杂证明可使得最坏时间复杂度为O(n^{3/2})。希尔排序的介绍到此为止,关于其他排序算法的介绍也会陆续更新,谢谢支持。

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





dreamcatcher-cx 关注 - 33

粉丝 - 1066

48

1

+加关注

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)
- 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)

5. 图解排序算法(二)之希尔排序(48)

«上一篇: 图解排序算法(一)之3种简单排序(选择,冒泡,直接插入)

» 下一篇: 谈谈Java中的ThreadLocal

posted @ 2016-11-27 17:34 dreamcatcher-cx 阅读(222686) 评论(27

) 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

国之动太	
MID MISS	•

· 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