

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 $O(n^2)$ 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。

基本思想

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，不管数组分布是怎麼样的，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。

公告

访问量：
88537

昵称：dreamcatcher-c
园龄：11个月
粉丝：45
关注：28
+加关注

| | | | | |
|----|--------|----|----|--|
| < | 2017年9 | | | |
| 日 | 一 | 二 | 三 | |
| 27 | 28 | 29 | 30 | |
| 3 | 4 | 5 | 6 | |
| 10 | 11 | 12 | 13 | |
| 17 | 18 | 19 | 20 | |
| 24 | 25 | 26 | 27 | |
| 1 | 2 | 3 | 4 | |

搜索

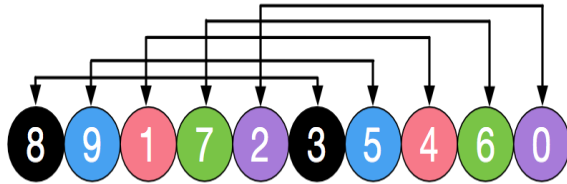
- 我的标签
- Oracle(3)
- hashmap(1)

- 随笔分类(20)
- java 基础
- java集合框架(1)
- jvm
- mysql

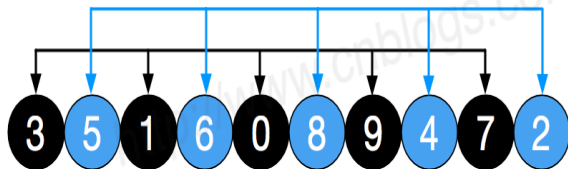
原始数组 以下数据元素颜色相同为一组



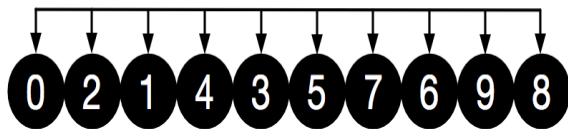
初始增量 $gap=length/2=5$, 意味着整个数组被分为5组, [8,3] [9,5] [1,4] [7,6] [2,0]



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:

```
1 package sortdemo;
2
3 import java.util.Arrays;
4
5 /**
6  * Created by chengxiao on 2016/11/24.
```

Oracle(4)

并发编程(8)

分布式系统

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)

积分与排名

积分 - 31271

排名 - 10249

最新评论

1. Re:HashMap实现原

@Paul_bai引用@dre
引用引用@Paul_bai谢
以推荐下, 谢谢! 好的

--d

2. Re:HashMap实现原

```
7  */
8  public class ShellSort {
9      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};
14          sort1(arr1);
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){
23          //增量gap，并逐步缩小增量
24          for(int gap=arr.length/2;gap>0;gap/=2){
25              //从第gap个元素，逐个对其所在组进行直接插入排序操作
26              for(int i=gap;i<arr.length;i++){
27                  int j = i;
28                  while(j-gap>=0 && arr[j]<arr[j-gap]){
29                      //插入排序采用交换法
30                      swap(arr,j,j-gap);
31                      j-=gap;
32                  }
33              }
34          }
35      }
36
37      /**
38       * 希尔排序 针对有序序列在插入时采用移动法。
39       * @param arr
40       */
41      public static void sort1(int []arr){
42          //增量gap，并逐步缩小增量
43          for(int gap=arr.length/2;gap>0;gap/=2){
44              //从第gap个元素，逐个对其所在组进行直接插入排序操作
45              for(int i=gap;i<arr.length;i++){
46                  int j = i;
47                  int temp = arr[j];
48                  if(arr[j]<arr[j-gap]){
49                      while(j-gap>=0 && temp<arr[j-gap]){
50                          //移动法
51                          arr[j] = arr[j-gap];
52                          j-=gap;
53                      }
54                      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       */
65      public static void swap(int []arr,int a,int b){
66          arr[a] = arr[a]+arr[b];
67          arr[b] = arr[a]-arr[b];
68          arr[a] = arr[a]-arr[b];
69      }
70 }
```

@dreamcatcher-cx引
谢不客气，有人才可以讨

3. Re:HashMap实现原

@Paul_bai谢谢...

--d

4. Re:Java原子类实现

@灯下轻敲恩恩，是的
引用类解决。不过说是
问题不太会影响并发准
到。感谢...

--d

5. Re:Java原子类实现

ABA漏洞可以采用java

阅读排行榜

- 1. HashMap实现原理及
4)
- 2. 图解排序算法(一)之
择，冒泡，直接插入)(1
- 3. Oracle体系结构详解
- 4. 图解排序算法(二)之
- 5. 图解排序算法(三)之

评论排行榜

- 1. HashMap实现原理及
- 2. 图解排序算法(二)之
- 3. 图解排序算法(三)之
- 4. 图解排序算法(五)之
取中法(4)
- 5. 图解排序算法(一)之
择，冒泡，直接插入)(2

推荐排行榜

总结

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{ $n/2, (n/2)/2 \dots 1$ } (希尔增量)，其最坏时间复杂度依然为 $O(n^2)$ ，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

好文要顶

关注我

收藏该文

dreamcatcher-cx

关注 - 28

粉丝 - 45

+加关注

3

推荐

0

反对

« 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(5077) 评论(6) 编辑 收藏

评论列表

#1楼 2016-11-28 10:46 p712long

那个8、9、1、7、2、3、5、4、6、0分组是不是有错？
分成5组分别是：[8,3],[9,5],[1,4],[7,6],[2,0]

支持(0) 反对(0)

#2楼[楼主] 2016-11-28 11:09 dreamcatcher-cx

@ p712long
调整的时候没注意到，已经修复了，谢谢提醒哈

支持(0) 反对(0)

#3楼 2016-11-28 13:07 p712long

这个希尔排序法：我不明白优点在哪里？对于非长数组排序，其实用JavaScript的sort和reverse一行代码就实现了：
var arr1 = [3, 1, 'd', 'b', 'c', 'a', 0, 9, 7].sort().reverse();
console.log(arr1.join(','));

支持(0) 反对(0)

#4楼[楼主] 2016-11-28 13:59 dreamcatcher-cx

@ p712long
你说的sort()函数是语言层面的对于排序算法的封装，我们现在讨论的希尔排序不过是排序操作的其中一种算法思想，其实每种语言类库里都有对于排序算法的封装，实际中我们大都直接调用就可以，很少自己去实现的（像Java中的Arrays.sort()）。js中的sort()函数，底层究竟使用了哪种排序算法，貌似不同浏览器会采用不同排序算法（快排，归并....）。

支持(0) 反对(0)

#5楼 2017-05-31 09:55 雾唤风月

问哈，第二种，移动法是不是有问题呀？为啥代码实现后，并没有完全排序呀。

我好像找到原因了，arr[j] = temp;这个应该放到while里面去吧？不然的话，就按照你的第二种方法的例子，跑出来的结果都不对。

支持(0) 反对(0)

1. HashMap实现原理

2. 谈谈Java中的volatile

3. Oracle体系结构详解

4. ConcurrentHashMap码分析(5)

5. 图解排序算法(一)之选择，冒泡，直接插入(4)

<http://www.cnblogs.com/chengxiao/p/6104371.html>

4/5

#6楼[楼主] 2017-05-31 15:10 dreamcatcher-cx

@ 雾唤风月
在对当前元素进行比较，并寻找最终位置进行插入的时候，判断条件应该是temp<arr[j-gap]，而不是arr[j]<arr[j-gap]。当然，您说的将arr[j]=temp放入while里面也可以，只不过这样就跟交换法一样了（每次比较后，若后值小于前值，都需要调整；移动法则是比较过程中，不符合排序规则的移动，然后找到当前元素的最终位置插入即可），我已经修改了，感谢指正。

支持(0) 反对(0)

刷新评论 刷新页面 返回顶部

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](#)。

- 最新IT新闻：
- 799元！360智能云镜S600亮相：6.86英寸大屏
 - 京东超市宣布：已是中国线上线下最大的超市
 - 武汉首现共享汽车肇事 使用人逃逸全赔并被记载征信记录
 - 腾讯音乐将出售3%股份
 - 暴风高管解读中期财报：争取魔镜明年盈利
- » 更多新闻...

- 最新知识库文章：
- 做到这一点，你也可以成为优秀的程序员
 - 写给立志做码农的大学生
 - 架构腐化之谜
 - 学会思考，而不只是编程
 - 编写Shell脚本的最佳实践
- » 更多知识库文章...