[返回主页](https://www.cnblogs.com/chengxiao/)

[dreamcatcher-cx](https://www.cnblogs.com/chengxiao/)

why is more important than what.

* [博客园](https://www.cnblogs.com/)
* [首页](https://www.cnblogs.com/chengxiao/)
* [新随笔](https://i.cnblogs.com/EditPosts.aspx?opt=1)
* [联系](https://msg.cnblogs.com/send/dreamcatcher-cx)
* [订阅](javascript:void(0))
* [管理](https://i.cnblogs.com/)

[图解排序算法(二)之希尔排序](https://www.cnblogs.com/chengxiao/p/6104371.html)

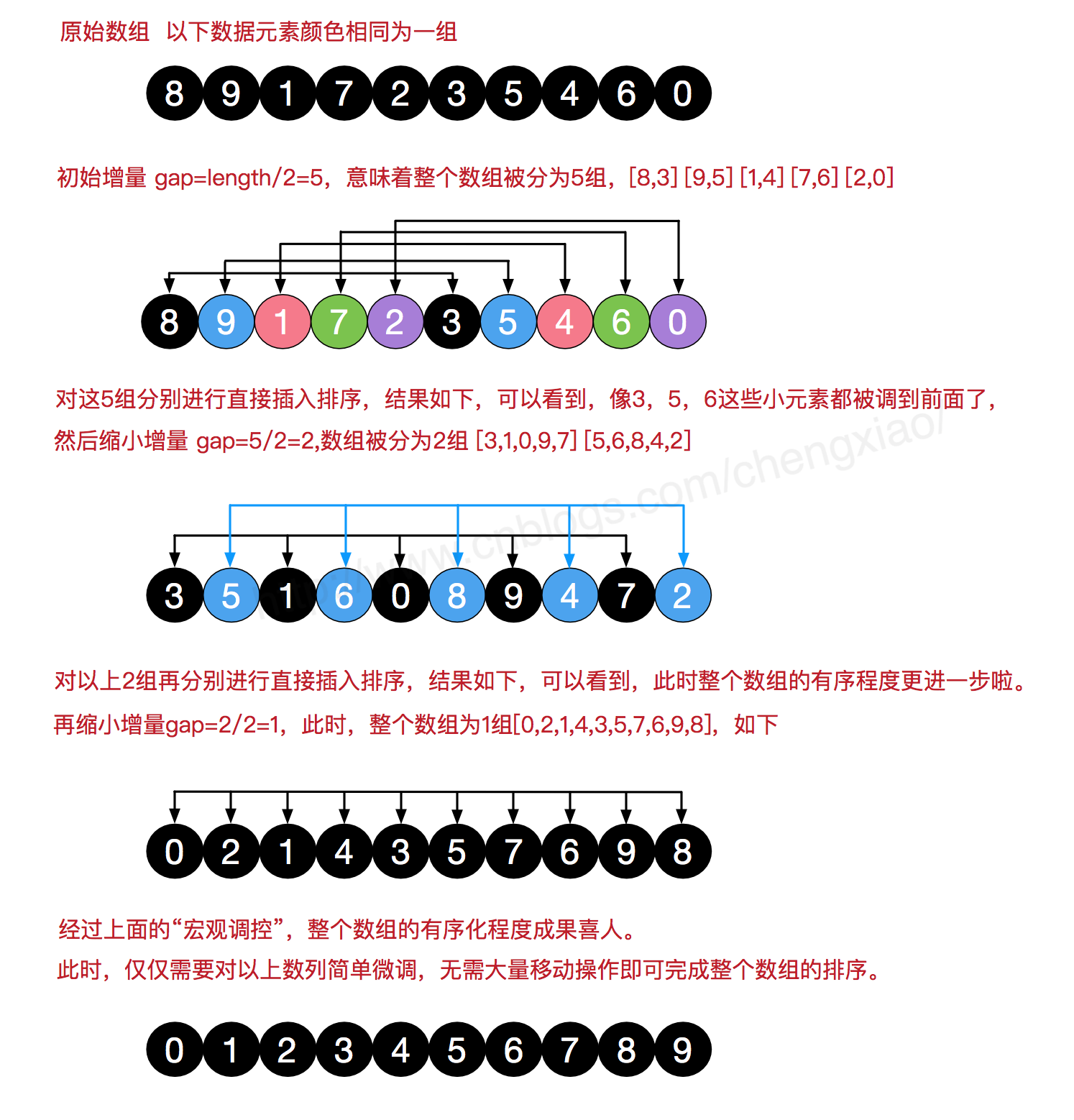
　　希尔排序是希尔（Donald Shell）于1959年提出的一种排序算法。希尔排序也是一种插入排序，它是简单插入排序经过改进之后的一个更高效的版本，也称为缩小增量排序，同时该算法是冲破O(n2）的第一批算法之一。本文会以图解的方式详细介绍希尔排序的基本思想及其代码实现。

**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。



**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：

[复制代码](javascript:void(0);)

1 package sortdemo;

2

3 import java.util.Arrays;

4

5 /\*\*

6 \* Created by chengxiao on 2016/11/24.

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 }

[复制代码](javascript:void(0);)

**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...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(n3/2)。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。