算法思路

希尔排序也是一种插入排序，它是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

在插入排序中，最好的情况是直接插入到排好序的末尾，最差的情况是排序到头部，那么所有元素向后移动一位

希尔排序优化思想在于，减少移动，要求不是那么严格。达到基本有序即可：小的关键字基本在前面，大的基本在后面，不大不小的基

文字说明

希尔排序是把记录按下表的一定增量分组（ji），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，组，算法便终止

希尔排序的基本步骤，在此我们选择增量gap=length/2（此增量不是最佳的，这里谨以此增量为说明该排序算法思路）

{8, 9, 1, 7, 2 ,3 ,5 , 4 ,6 ,0}

第一轮 增量为 length / 2 = 5

- 第一个元素是8，增量5 就是第六个元素 3，在增量5 超出了数组长度，那么第一组就是 8 和 3
- 第二个元素是9，增量5 就是第七个元素 5，在增量5 超出了数组长度，那么第二组就是 9 和 5
- 分组后如下：{8,3} ，{9，5}，{1，4}，{7，6}，{2，0}
- 对上面5组分别进行插入排序后数组结果：{3，5，1，6，0，8，9，4，7，2}

第二轮 增量为 上轮增量的一半 也就是 5/2 = 2

- 第一个元素是3，增量是2，那么这一组第二个元素是1，继续增量2 ，第三个元素是0，依次类推，直到超出长度
- 分组后如下：{3，1，0，9，7} 、{5，6，8，4，2}
- 对上面2组分别进行插入排序后数组结果：{0，2，1，4，3，5，7，6，9，8}

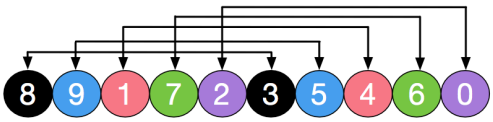
第三轮 增量为 上一轮的一半，也就是 2/2 = 1

- 将后面的结果进行最后一次插入排序
- {0，1，2，3，4，5，6，7，8，9}

图解说明

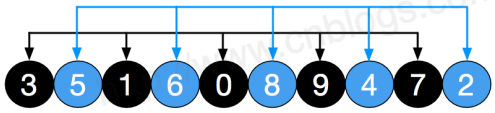
原始数组 以下数据元素颜色相同为一组

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]



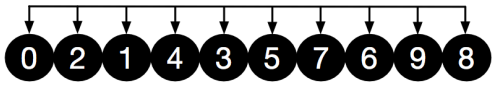
对这5组分别进行直接插入排序，结果如下，可以看到，像3，5，6这些小元素都被调到前面了，

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

```java
package day11.排序;

/**
 * 希尔排序
 */
public class ShellSort {

    public static void main(String[] args) {
        int[] array = {8, 9, 1, 7, 2, 3, 5, 4, 6, 0};

        //未排序数组顺序为
        System.out.println("未排序数组顺序为: ");
        display(array);
        System.out.println("------------------------");
        shellSort(array);
        System.out.println("经过排序后的数组顺序为: ");
        display(array);

    }

```

```
21      private static void shellSort(int[] array) {
22          int len = array.length;
23          int temp, gap = len / 2;
24
25          //如果 gap = 1 , 1 / 2 = 0
26          while (gap > 0) {
27              for (int i = gap; i < len; i++) {
28                  temp = array[i];
29                  int preIndex = i - gap;
30                  while (preIndex >= 0 && array[preIndex] > temp) {
31                      array[preIndex + gap] = array[preIndex];
32                      preIndex -= gap;
33                  }
34                  array[preIndex + gap] = temp;
35              }
36              gap /= 2;
37          }
38      }
39
40
41      public static void display(int[] array) {
42          for (int i = 0; i < array.length; i++) {
43              System.out.print(array[i] + " ");
44          }
45          System.out.println();
46      }
47  }
```

总结

希尔排序的实质就是分组插入排序，又称缩小增量法。

将整个无序序列分割成若干个子序列（由相隔某个“增量”的元素组成的）分别进行直接插入排序，然后依次缩减增量再进行排序，待整元素进行一次直接插入排序。

因为直接插入排序在元素基本有序的情况下，效率是很高的，因此希尔排序在时间效率上有很大提高