**排序总结**

**排序算法稳定性**

假设在数列中存在a[i]=a[j]，若在排序之前，a[i]在a[j]前面；并且排序之后，a[i]仍然在a[j]前面。则这个排序算法是稳定的！



**一、冒泡排序**

**基本思想：**

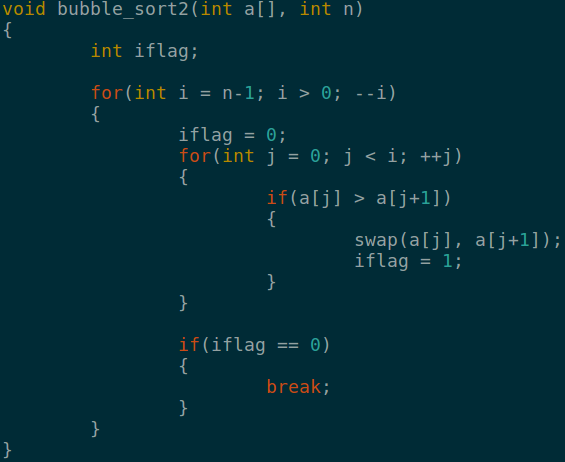
遍历若干次要排序的数列，每次遍历时，它都会从前往后依次的比较相邻两个数的大小； 如果前者比后者大，则交换它们的位置。这样，一次遍历后，最大的元素就在数列的末尾！

添加一个标记，如果一趟遍历中发生了交换，标记置为true，否则为false。如果某一趟遍历中没发生交换，就说明排序已经完成！

**时间复杂度：**0(N2)

**算法稳定性：**稳定

Screenshot from 2018-11-06 10-00-18



**二、插入排序**

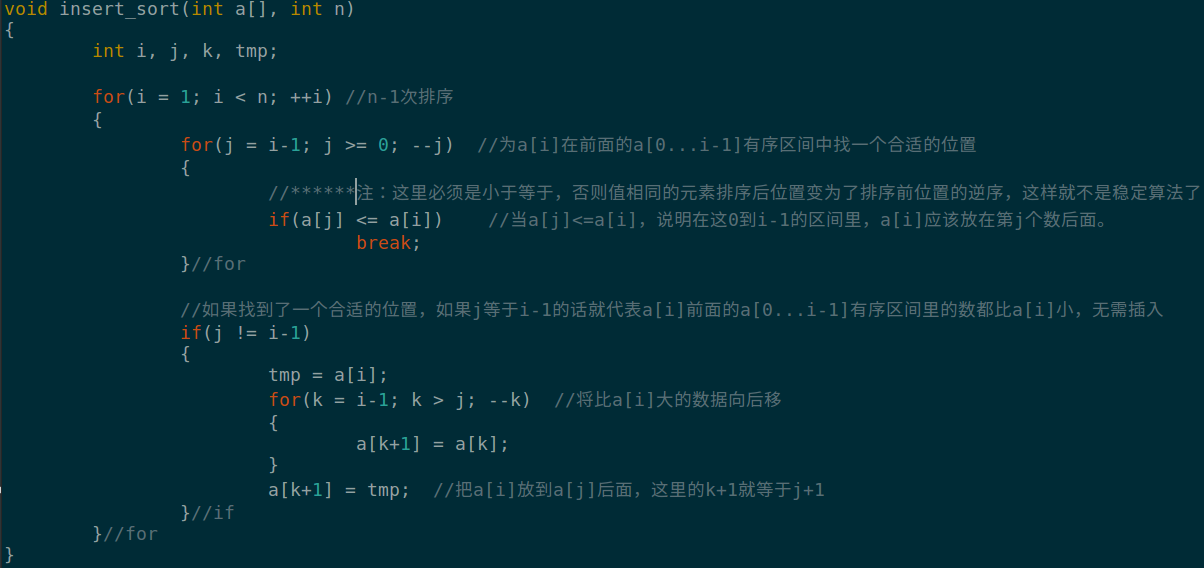
**1、直接插入排序**

**基本思想：**

把n个待排序的元素看成为一个有序表和一个无序表。开始时有序表中只包含1个元素，无序表中包含有n-1个元素，排序过程中每次从无序表中取出第一个元素，将它插入到有序表中的适当位置，使之成为新的有序表，重复n-1次可完成排序过程。

**时间复杂度：**O(N2)

**算法稳定性：**稳定



算法解释：从数列中第2个数开始a2，直到最后一个数，总共n-1次，每次在ai前面的i-1个数中从第i-1个数开始找，找到第一个aj（0<j<i）比ai小的数，一旦找到，把j+1到i-1这些个数向后移一个，然后把ai放到j+1的位置。

**注：**在查找那个a[j]的时候必须用的是小于等于！否则值相同的元素排序后位置变为了排序前位置的逆序，这样就不是稳定算法了

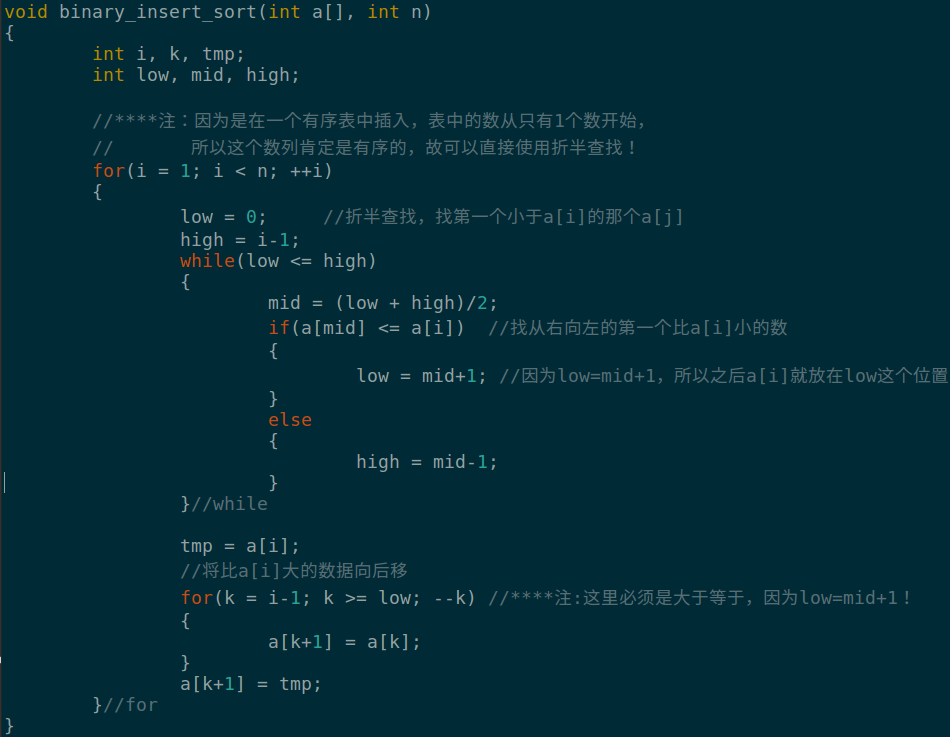
**2、折半插入排序**

**基本思想：**

把n个待排序的元素看成为一个有序表和一个无序表。开始时有序表中只包含1个元素，无序表中包含有n-1个元素，排序过程中每次从无序表中取出第一个元素，将它插入到有序表中的适当位置（查找这个位置的方法是采用折半查找的方法，即在a[0...i-1]这个有序区间里通过折半查找来满足a[j]<a[i]的那个a[j]），使之成为新的有序表，重复n-1次可完成排序过程。

当待排序序列中的记录数量n很大时，折半插入排序进行的关键字间的比较次数比直接插入排序少！

**时间复杂度：**O(N2)



1. **希尔排序(shell)**

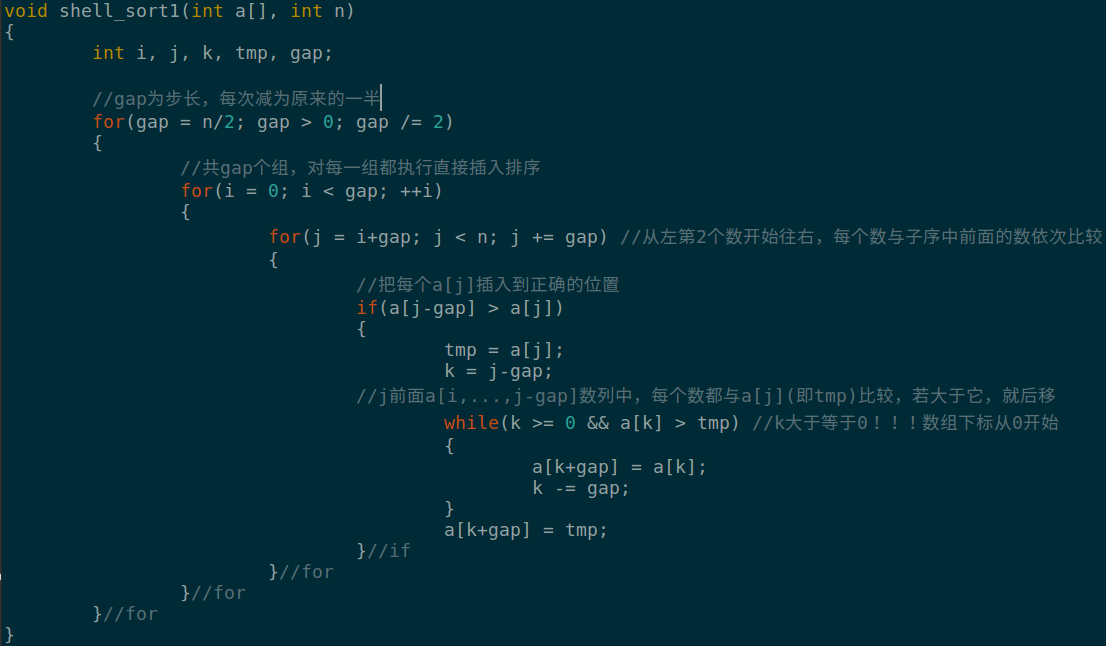
希尔排序(Shell Sort)是插入排序的一种，它是针对直接插入排序算法的改进，该方法又称缩小增量排序。希尔排序实质上是一种分组插入方法

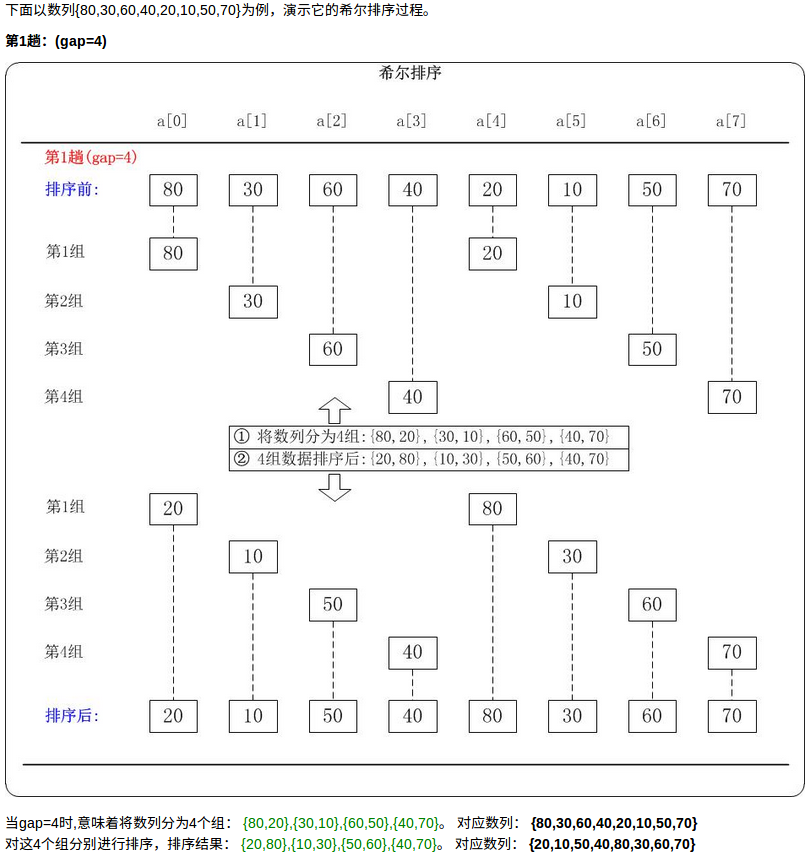
**基本思想：**

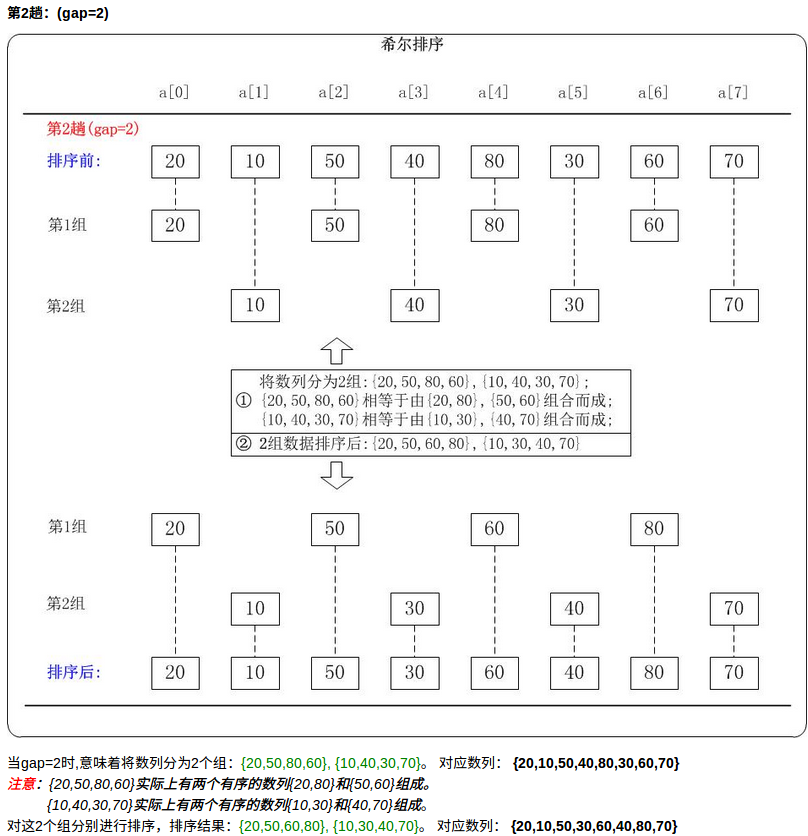
对于n个待排序的数列，取一个小于n的整数gap(gap被称为步长)将待排序元素分成若干个组子序列，所有距离为gap的倍数的记录放在同一个组中。然后，对各组内的元素进行直接插入排序。这一趟排序完成之后，每一个组的元素都是有序的。然后减小gap的值，并重复执行上述的分组和排序。重复这样的操作，当gap=1时，整个数列就是有序的。

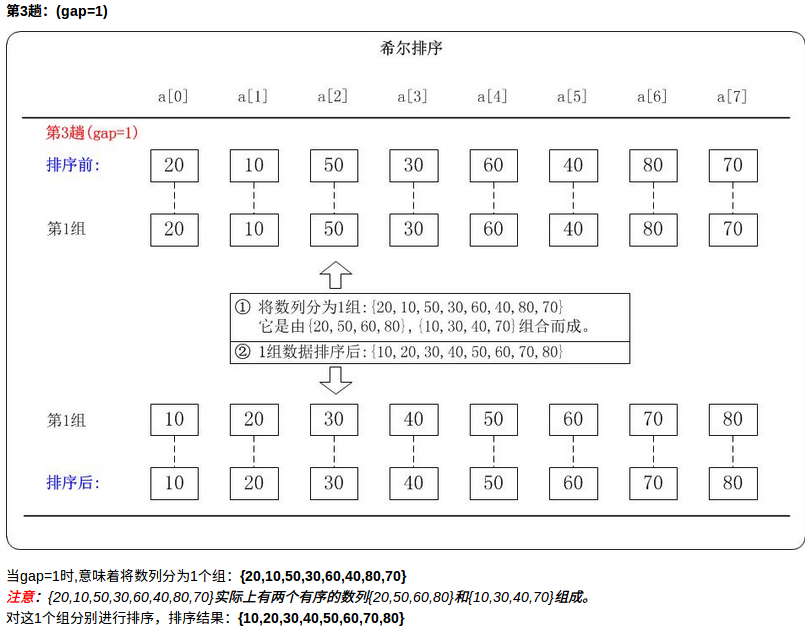
**时间复杂度：**希尔排序的时间复杂度与增量(即，步长gap)的选取有关。例如，当增量为1 时，希尔排序退化成了直接插入排序，此时的时间复杂度为O(N²)，而Hibbard 增量的希尔排序的时间复杂度为O(N3/2)。

**算法稳定性：**不稳定，对于相同的两个数，可能由于分在不同的组中而导致它们的顺序发生 变化。









**三、快速排序**

**基本思想：**

把n个待排序的元素看成为一个有序表和一个无序表。开始时有序表中只包含1个元素，无序表中包含有n-1个元素，排序过程中每次从无序表中取出第一个元素，将它插入到有序表中的适当位置，使之成为新的有序表，重复n-1次可完成排序过程。

**时间复杂度：**O(N2)

**算法稳定性：**稳定