**一、冒泡排序**

基本思想是:两两比较相邻记录的关键字,如果反序则交换。

**二、直接插入排序**

将一个记录插入到已经排好序的有序表中, 从而得到一个新的,记录数增1的有序表。

**三、简单选择排序**

通过n-i次关键字之间的比较,从n-i+1 个记录中选择关键字最小的记录,并和第i(1<=i<=n)个记录交换之。

**四、希尔排序**

先将整个待排元素序列分割成若干子序列（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，待整个序列中的元素基本有序（增量足够小）时，再对全体元素进行一次直接插入排序（增量为1）。

**五、归并排序**

假设初始序列含有n个记录,则可以看成n个有序的子序列,每个子序列的长度为1,然后两两归并,得到(不小于n/2的最小整数)个长度为2

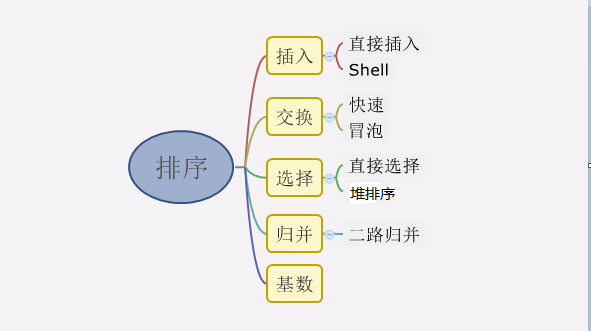
或1的有序子序列,再两两归并,...如此重复,直至得到一个长度为n的有序序列为止,这种排序方法称为2路归并排序。

**六、堆排序**

堆是具有下列性质的完全二叉树:每个节点的值都大于或等于其左右孩子节点的值,称为大顶堆；或者每个节点的值都小于或等于其左右孩子节点的值,称为小顶堆。

**七、快速排序**

通过一趟排序将要排序的数据分割成独立的两部分，其中一部分的所有数据都比另外一部分的所有数据都要小，然后再按此方法对这两部分数据分别进行快速排序，整个排序过程可以递归进行，以此达到整个数据变成有序序列。



时间复杂度：预计算法的运算时间。

稳定的：若存在多个关键字相同记录，排序后，相同关键字的记录之间的相对次序保持不变，则称其为稳定的。

不稳定的：若具有相同关键字的记录之间的相对次序发生变化，则称其为不稳定的。

**1.插入排序**

**1.1.直接插入**

基本思想：

　　依次将每个记录插入到一个已排好的有序表中去，从而得到一个新的、记录数增加1的有序表。其中有i个记录，记录R[0]有两个作用：一是防止丢失指针指向的值，二是防止越界，不用判断循环条件。



时间复杂度：

　最好的情况：正序有序，这样只需要比较N次，不需要移动，所以时间复杂度为O(n)

　最坏的情况：逆序有序，这样每一个数需要比较N次，一共有N个元素，所以时间复杂度为O（n­2）

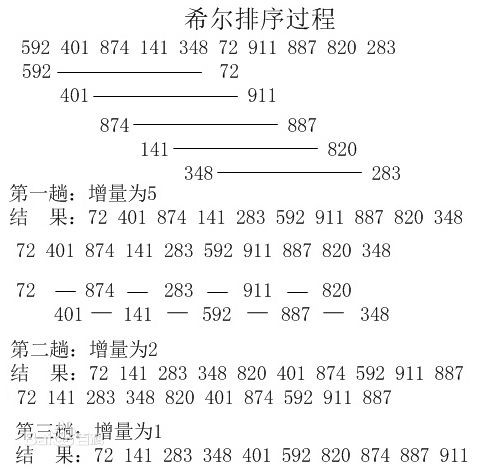
　平均情况：O(n­2)

稳定性：

　　是稳定的。直接插入排序是一个有序表和一个无序表，如果A和B是两个相同的数字，那么当A和B进行比较的时候，直接插入到A的后边，没有必要去移动位置，所以是稳定的。

**1.2.Shell排序**

　　基本思想：缩小增量排序，在直接插入排序上的一个改进。每趟都按照确定的间隔将元素进行分组，在每一组中进行直接插入排序，使得小的数可以跳跃式的前进。然后逐步的将步长缩小，直到步长为1...等到序列基本有序，进行直接插入排序。



时间复杂度：

　最好情况：shell排序的好坏与步长的选择有关系，目前不知道最好的步长如何选，所以不知道最好的时间复杂度。

　最坏情况：O(N\*logN)

　平均情况：O(N\*logN)

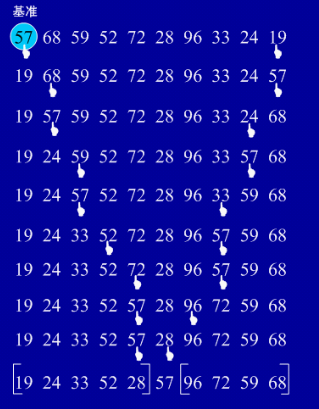
稳定性：

　　不稳定的。相同的数会在自己的组内进行插入排序的移动，因此稳定性就会被破坏。

**2.交换排序**

**2.1.快速排序**

　　基本思想：采用分治法，把大的问题进行分解。在N个记录中去某一个记录的键值为标准，通常取第一个记录键值为标准，通过一趟排序将待排的记录分为小于或等于这个键值和大于这个键值的两个独立的部分，这时一部分的记录键值均比另一部分小，然后对着两部分记录继续分别进行快速排序，已达到整个序列的有序。



时间复杂度：

　最好情况：每次都将序列分为两个部分，故为 O(N\*log2N)

　最坏情况：基本有序时，退化为冒泡排序，几乎要比较N\*N次，故为O(N\*N)

　平均情况： O(N\*log2N)

稳定性：

　　不稳定。每次需要和基准交换，原来顺序一定会被打乱。

**2.2.冒泡排序**

基本思想：采用从最后一个数据依次向前比较，找出最小的一个来。每次都是和相邻的交换位置。



时间复杂度：

　最好情况：正序有序，则只需要比较n次。故，为O(n)

　最坏情况：逆序有序，则需要比较(n-1)+(n-2)+……+1，故，为O(N^2)

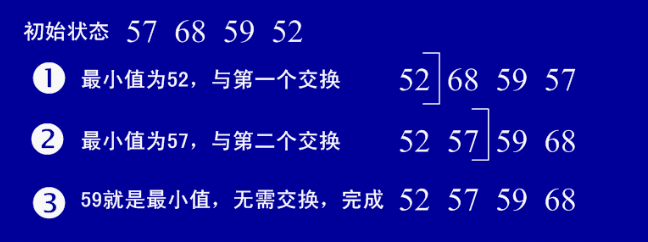
　平均情况：(N^2)

稳定性：稳定。排序过程中只交换相邻两个元素的位置。

**3.选择排序**

**3.1.直接选择**

　　基本思想：第一次从R[0]~R[n-1]中选取最小值，与R[0]交换，第二次从R[1]~R[n-1]中选取最小值，与R[1]交换，....，第i次从R[i-1]~R[n-1]中选取最小值，与R[i-1]交换，.....，第n-1次从R[n-2]~R[n-1]中选取最小值，与R[n-2]交换，总共通过n-1次，得到一个按排序码从小到大排列的有序序列



时间复杂度：

　最好情况：交换0次，但是每次都要找到最小的，遍历N\*N次，因此为O(N\*N)。

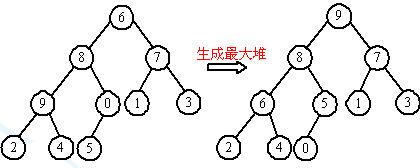
　最坏情况：O(N\*N)

　平均情况：O(N\*N)

稳定性：不稳定。相同的两个数会跨距离的进行交换。

**3.2.堆排序**

　　基本思想：利用大顶锥的构建，使在无序记录中找最大值变得简单。



时间复杂度：

　最好情况：O(N\*log2N)

　最坏情况：O(N\*log2N)

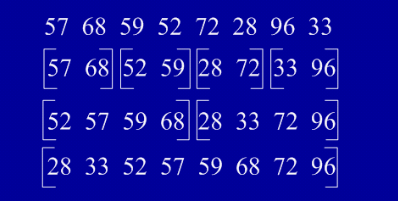
　平均情况：O(N\*log2N)

稳定性：不稳定。堆需要不断的进行调整。

**4.归并**

**4.1.二路归并**

　　基本思想：将两个有序表合成一个有序表。



时间复杂度：

　最好情况：O(N\*log2N)

　最坏情况：O(N\*log2N)

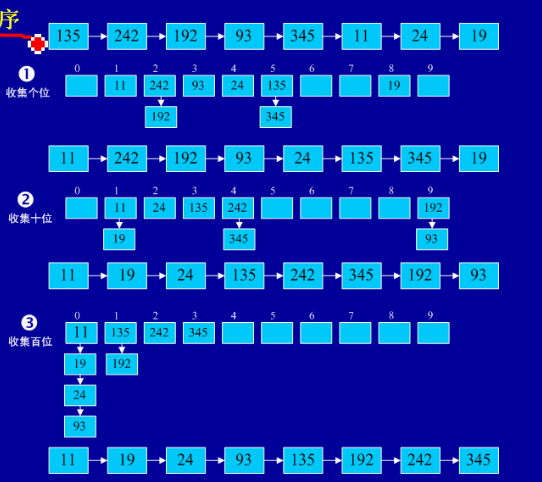
　平均情况：O(N\*log2N)

稳定性：稳定。不会改变元素的相对位置。

**5.基数**

**5.1.基数排序**

　　基本思想：很简单不用比较的，按照个位排序，再按照十位排序，最后按照百位排序。



时间复杂度：

　最好情况：O(d(r+n))

　最坏情况：O(d(r+n))

　平均情况：O(d(r+n))

稳定性：稳定。不会改变元素的相对位置。

**6.总结**



   还需理解才能灵活掌握，关于时间复杂度还需要深入的去理解。