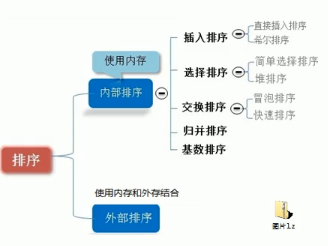
# 数据结构与算法

#算法思想：由简单到难，把复杂算法拆分乘简单问题逐步解决



**#基数排序就是桶排序的扩展**

内部排序：

指将需要处理的所有数据都加载到内部存储器中进行排序

外部排序：

数据量过大，无法全部加载到内存中，需要借助外部存储进行排序。

度量一个程序（算法）执行时间的两种方法：

**1.事后统计法**

面临两问题：需要**实际运行**程序；所得时间依赖于计算机硬件、软件等**环境因素**

#且这种方式，要在**同一台计算机的相同状态**下运行，才能比较哪个算法速度更快。

**2.事前估算法**

直接分析某个算法的**时间复杂度**来判断哪个算法更优

**时间频度：**

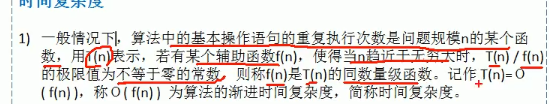
一个算法中的语句执行次数称为语句频度或时间频度，记为T（n）。

#忽略常数项，即：3n+10，3n可以看作他们的时间复杂度为3n。（10被省略）

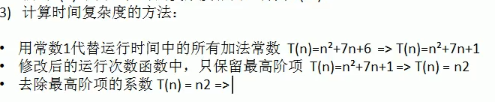
#忽略低次项，即：2n^2+3n+10，2n^2可以看作为2n^2。（忽略3n+10）

#忽略系数，即：5n^2+7n，3n^2+2n可以看作n^2。（5，3忽略，低次项忽略）

**时间复杂度：**

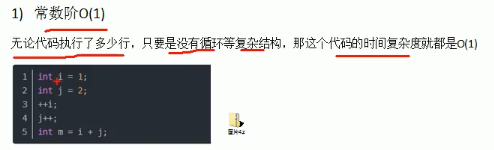


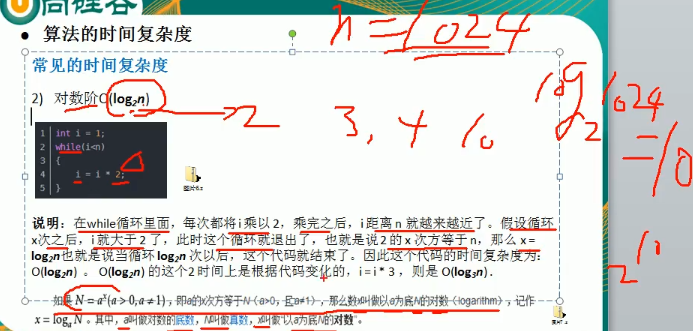


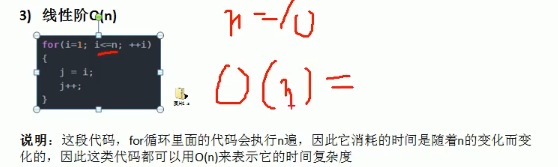




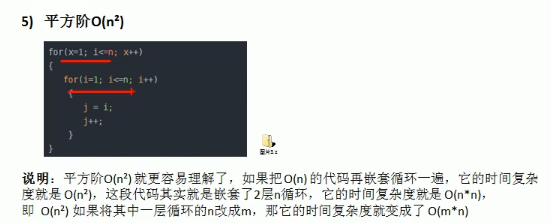
**（按时间复杂度往下递增，时间复杂度越大，算法效率执行效率越低，避免指数阶）**

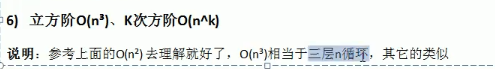










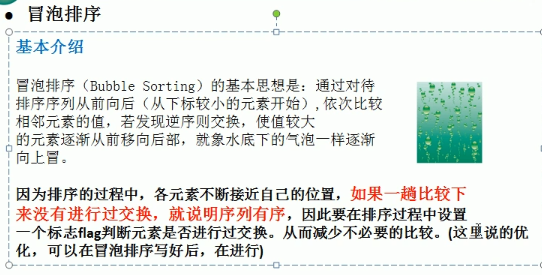


**平均时间复杂度、最坏时间复杂度：**

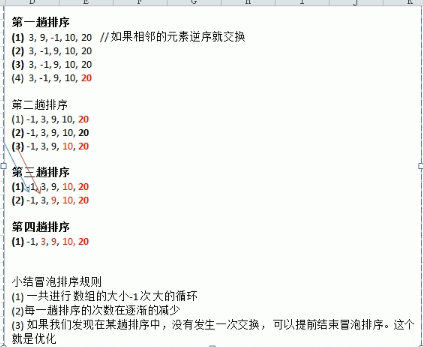




**1.冒泡排序：（每次找到一个最大数放后面，外：i<n-1，内：j<n-1-i，i为趟数）**



图解：

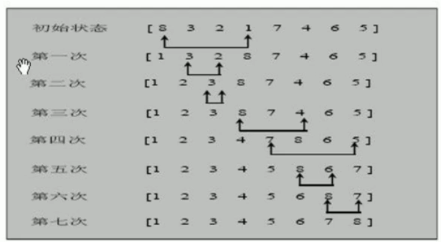


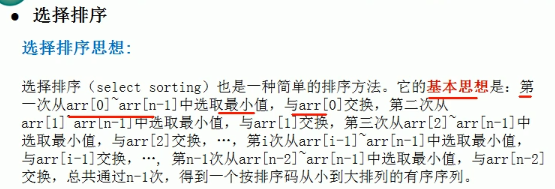


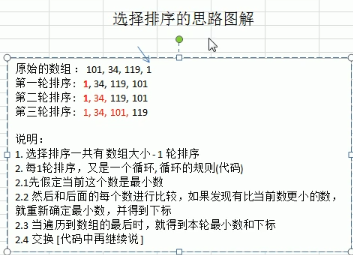
**2.选择排序：（选择排序每次遍历交换1次，冒泡每次遍历都交换多次）**

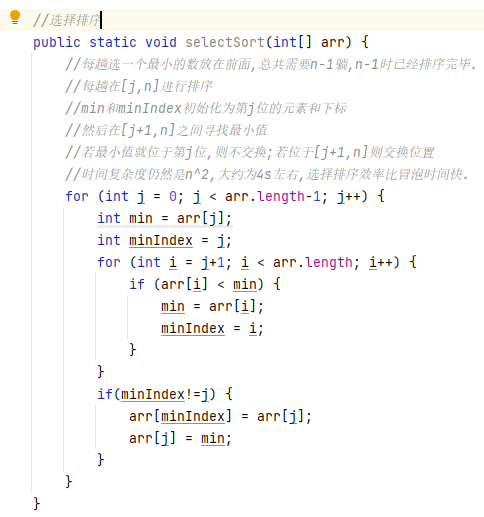
**（选择排序优于冒泡排序）**

***//每趟选一个最小的数放在前面,总共需要n-1躺,n-1时已经排序完毕.  
//每趟在[j,n]进行排序  
//min和minIndex初始化为第j位的元素和下标  
//然后在[j+1,n]之间寻找最小值  
//若最小值就位于第j位,则不交换;若位于[j+1,n]则交换位置  
//时间复杂度仍然是n^2，时间大概4s左右（8w条数据）***





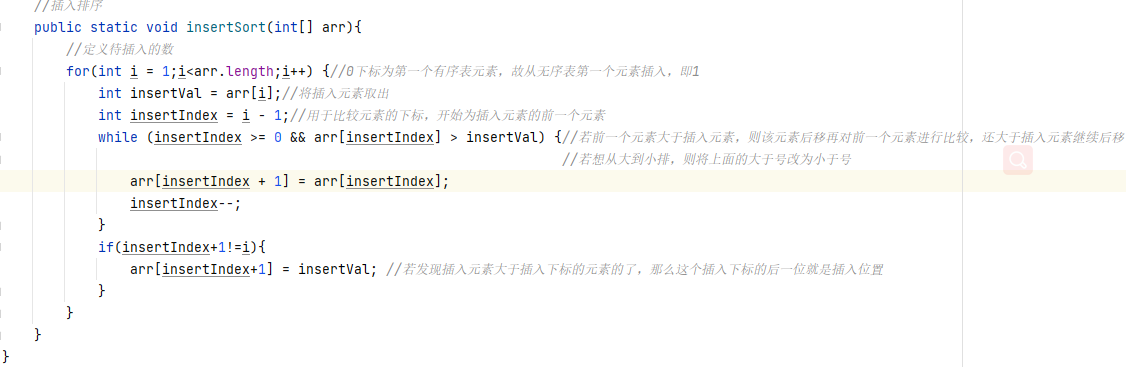




**3.插入排序（直接插入排序）**

**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一个元素，无序表中包含n-1个元素。排序过程每次从无序表中取出第一个元素，把它的排序码依次与有序表元素的排序码进行比较，将它插入到有序表中的适当位置，使之称为新的有序表。**

**存在问题：倘若插入的数最小，那么后移次数明显增大多，影响效率**



**4.希尔排序**

**直接插入排序的优化版本，也称为缩小增量排序。**

**把数组按下标的一定增来量分组，对每组使用直接插入排序算法，随着增量逐渐减少，有一个增量序列，例如[1,3,5]**

**步骤：**

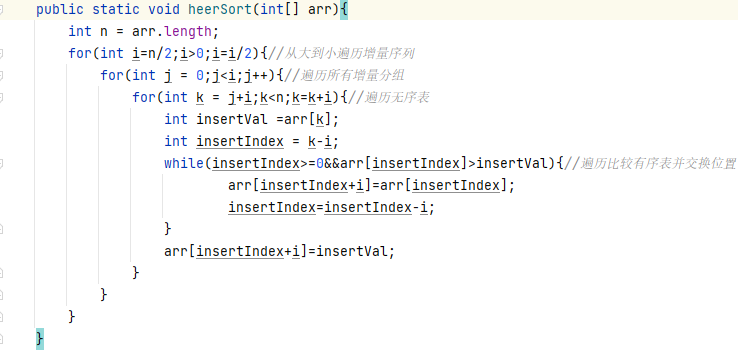
**1.从大到小遍历增量序列**

**2.遍历增量分组**

**3.遍历无序表**

**4.遍历有序表并比较插入值与有序表元素**

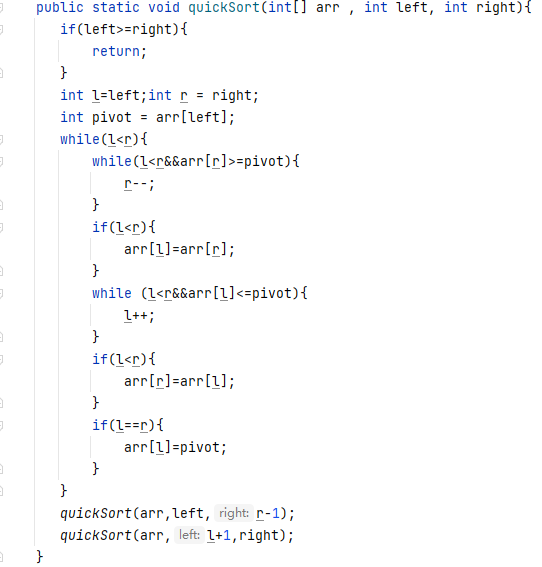
**5.交换位置**



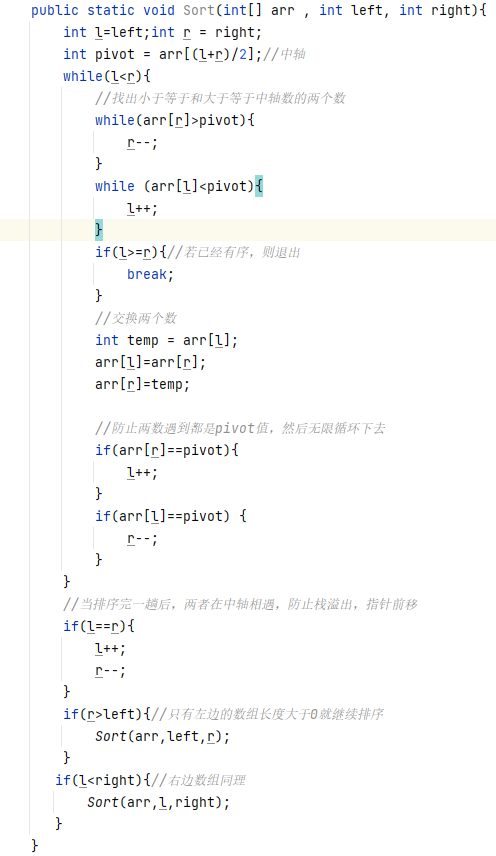
**5.快速排序**

**冒泡排序的改进，基本思想：通过一趟排序将要排序的数据分割撑独立的两部分，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，整个排序可以递归进行，以此达到整个数据变成有序序列。**

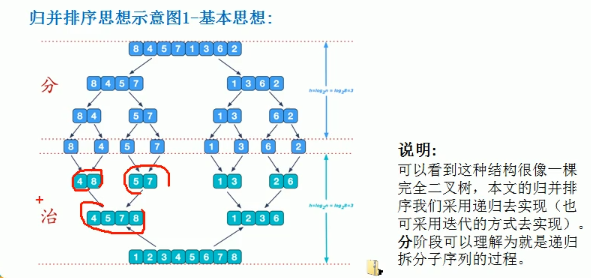
**交替插入法：**

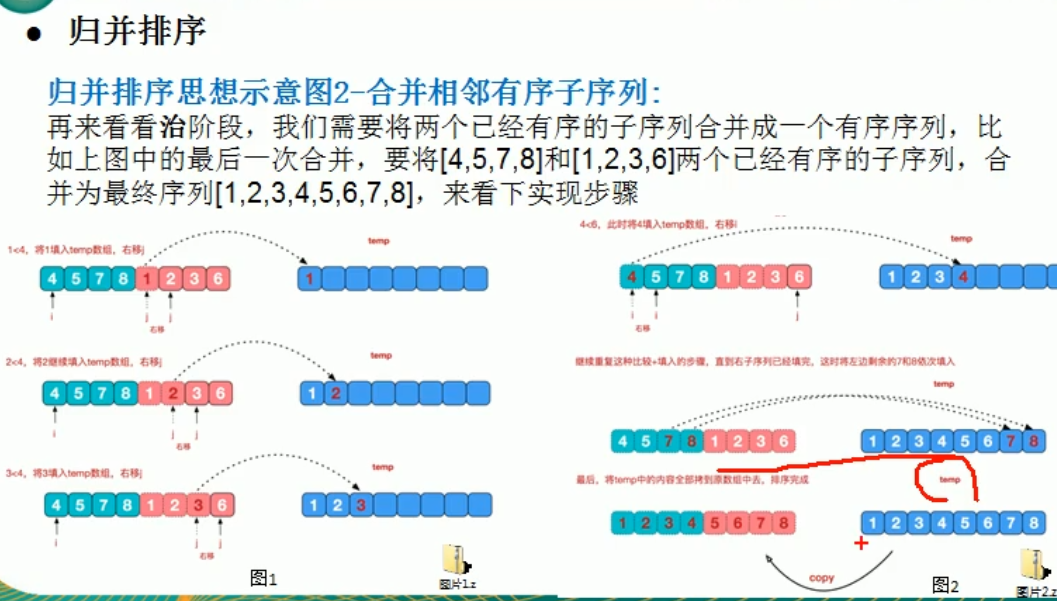


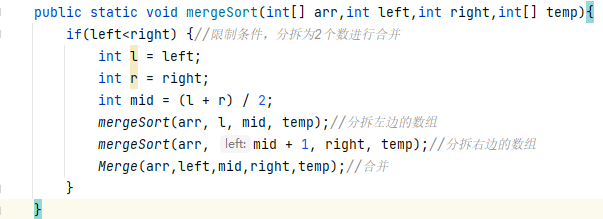
**两数交换法：**



**6.归并排序**





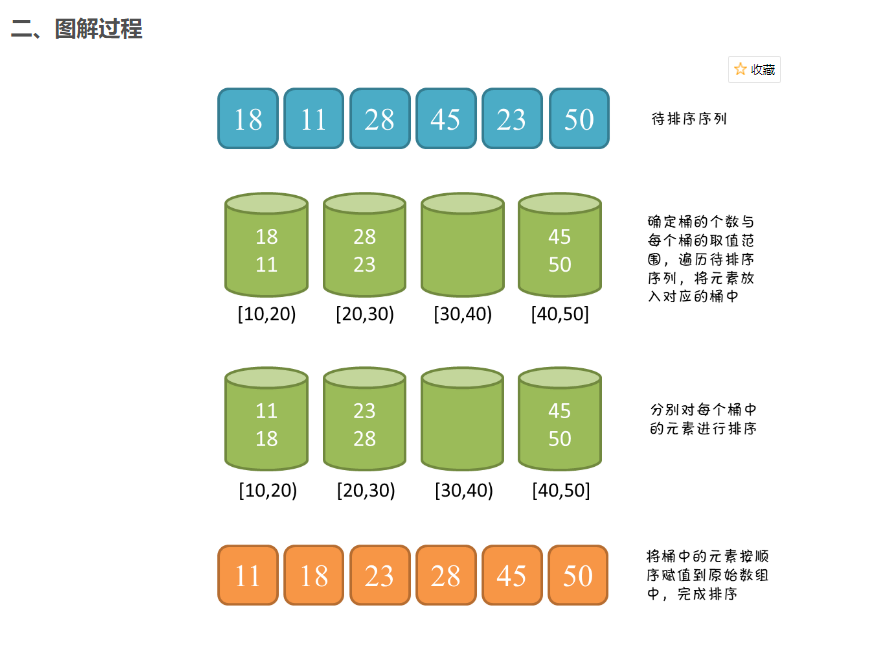




**7.基数排序**



**8.桶排序**



**9.计数排序**





10.堆排序