# 排序-插入&希尔&归并

你所知道的排序算法有哪些？

快速排序、冒泡排序，希尔排序，二分排序（二路归并）(nlogn)，桶排序，堆排序，基数排序，插入O（n^2），选择排序。

插入&希尔&归并排序：递进

选择&冒泡&快速：递进

堆排序：树论高级篇里面。

我们通常从哪几个方面来分析一个排序算法？

1.时间效率：决定了算法运行多久，O（1）

2.空间复杂度：

3.比较次数&交换次数:排序肯定会牵涉到两个操作，一个比较是肯定的。交换。

4.稳定性：这是什么？

1 9 3 5 3

第一种：1 3 3 5 9

第二种：1 3 3 5 9

哪一种是稳定的？相同的两个数排完序后，相对位置不变。

稳定排序有什么意义？应用在哪里呢？

电商里面订单排序：首先会按金额从小到大排，金额相同的按下单时间。我从订单中心过来的时候已经按照时间排好序了。

1 8:01 65

2 20:05 30

3 21:10 30

4 22:01 45

我选择排序算法：如果我选择不稳定的排序算法 那我还要比较两次的，如果我选择稳定的排序算法 那我就只要比较一个字段。

假设有个这样的问题：打扑克。分成两部分：一部分是你手里的牌（已经排好序），一部分是要拿的牌（无序）。把一个无序的数列一个个插入到有序数列中。

一个有序的数组，我们往里面添加一个新的数据后，如何继续保持数据有序呢？我们只要遍历数组，找到数据应该插入的位置将其插入即可。

画图演示：

以上这种往一个有序的集合里面插入元素，插入后序列仍然有序这就是插入排序算法思路。理解起来不难吧。那么插入排序具体是怎么实现呢？

具体步骤如下：

1.将数组分成已排序段和未排序段。初始化时已排序端只有一个元素

2.到未排序段取元素插入到已排序段，并保证插入后仍然有序

3.重复执行上述操作，直到未排序段元素全部加完。

有几种数据结构，用什么数据结构来实现。数组，链表

看以下这个例子：对7 8 9 0 4 3进行插入排序

7 8 9 0 4 3

7 8 9 0 4 3

7 8 9 0 4 3

0 7 8 9 4 3

0 4 7 8 9 3

0 3 4 7 8 9

从以上操作中我们看到插入排序会经历一个元素的比较以及元素的移动。当我们从待排序列中取一个数插入到已排序区间时，需要拿它与已排序区间的数依次进行比较，直到找到合适的位置，然后还要将插入点之后的元素进行往后移动。

那么用代码怎么实现呢？

int a[] = { 9, 8, 7, 0, 1, 3, 2 };

int n = a.length;

//这里面会有几层循环 2

//时间复杂度：n^2

//最好的情况：什么情况下：O(n); O(1)

for(int i = 1 ; i < n;i++){ //为什么i要从1开始？ 第一个不用排序，我们就把数组从i分开，0~I的认为已经排好序

int data = a[i];

int j = i -1;

for(;j>=0;j--){//从尾到头 1+2+3+4+5+...+n=>

if(a[j] > data){

a[j+1] = a[j]; // 数据往后移动

}else{ //因为前面已经是排好序的 那么找到一个比他小的就不用找了，因为前面的肯定更小

break; //O(1) 如果这个break执行的越多 那么我是不是效率就越高?

}

}

a[j+1] = data;

System.out.print("第" +i +"次的排序结果为：");

for(j = 0 ; j < n;j++){

System.out.print(a[j]+" ");

}

System.out.println();

}

插入排序最好的情况下复杂度为O(n) 每次都走break

如果把数组分成小段，先每小段排序，两个小段合并的时候会走更多次的break

希尔排序其实是插入排序的一个改进版。他是怎么改进呢？

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。

先取一个小于n的整数d1作为第一个增量，把文件的全部记录分组。所有距离为d1的倍数的记录放在同一个组中。先在各组内进行直接插入排序；然后，取第二个增量d2<d1重复上述的分组和排序，直至所取的增量 =1( < …<d2<d1)，即所有记录放在同一组中进行直接插入排序为止。

其实就是分成很多小组使序列尽可能的变成有序段，因为我们通过对插入排序分析可知，插入排序对已经排好序的序列速度是很快的。

来看一个具体的过程：7 8 9 0 4 3 1 2 5 10

我们取的这个增量分别就是5 2 1

来看一个具体的过程：

按照一个增量分段：add=n/2 n=10 =>5,2,1

7 8 9 0 4 3 1 2 5 10

我们取的这个增量分别就是5 2 1

7 8 9 0 4 3 1 2 5 10:分出来的数组元素都是一样的

完成一次排序：

3 1 2 0 4 7 8 9 5 10

3 2 4 8 5：取增量为2的分为一组了

3 1 2 0 4 7 8 9 5 10

完成一次排序：

2 0 3 1 4 7 5 9 8 10

最后一次我们就取增量为1的分组：

2 0 3 1 4 7 5 9 8 10

完成一次排序：

0 1 2 3 4 5 7 9 8 10

就是一个完整的序列