软件设计师
--数据结构基础

高级项目经理　任铄
QQ：1530841586

第一章 数据结构基础

- 1.1 线性表
- 1.2 树和二叉树
- 1.3 图
- 1.4 排序
- 1.5 查找

向上人生路！

一、排序

将一组杂乱无章的数据按一定的规律次序排列起来。

排序的目的是什么？

高级项目经理　任钭
QQ：1530841586

- 便于查找！

排序算法的好坏如何衡量？

- 时间效率——排序速度（即排序所花费的全部比较次数）
- 空间效率——占内存辅助空间的大小
- 稳定性——若两个记录A和B的关键字值相等，但排序后A、B的先后次序保持不变，则称这种排序算法是稳定的。

排序前 (56, 34, 47, 23, 66, 18, 82, 47)

- 若排序后得到结果

  (18, 23, 34, 47, 47, 56, 66, 82)

  则称该排序方法是稳定的

- 若排序后得到结果

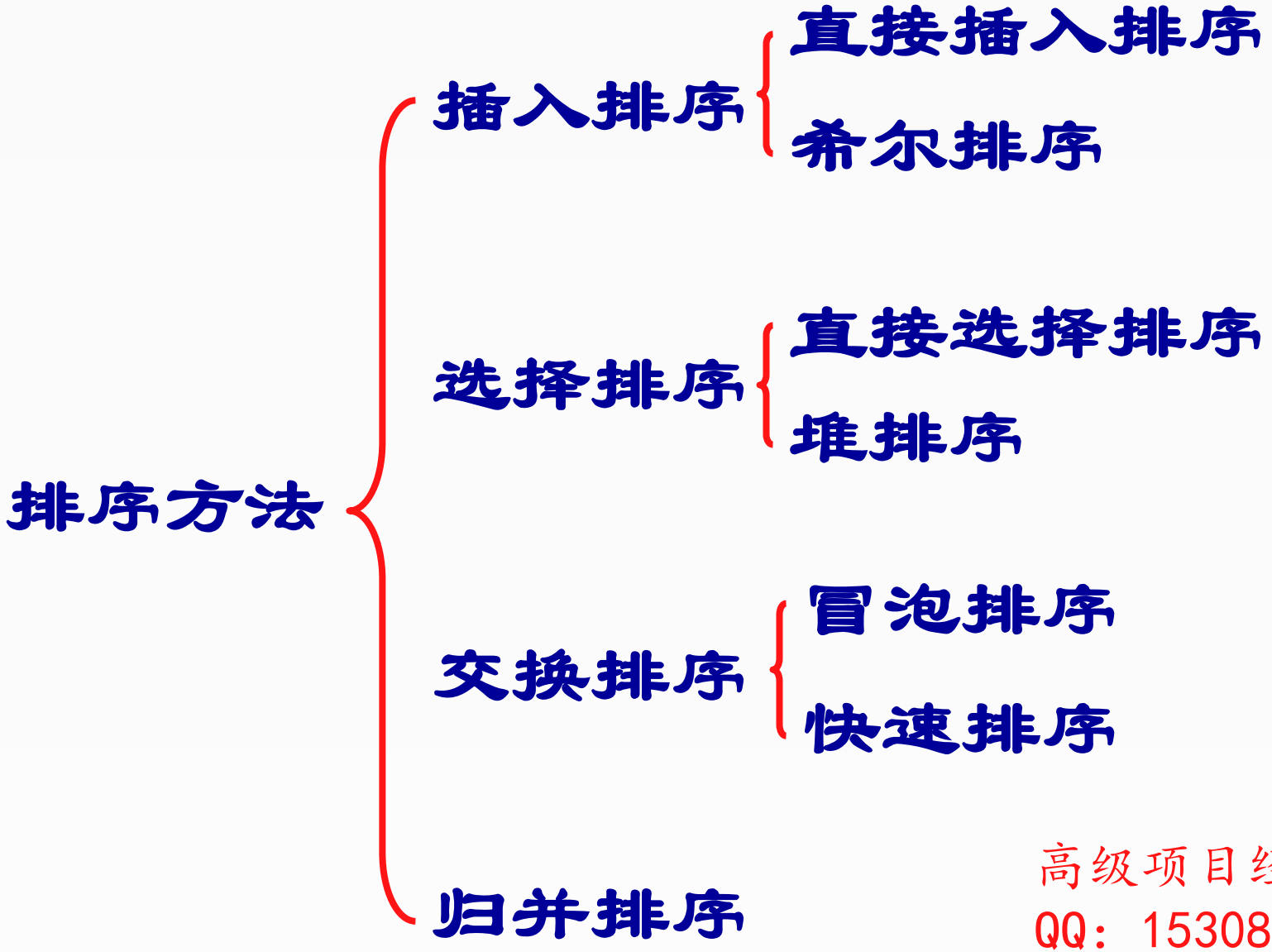
  (18, 23, 34, 47, 47, 56, 66, 82)

  则称该排序方法是不稳定的

高级项目经理　任钭
QQ：1530841586

- 内部排序：指待排序记录全部存放在内存中排序的过程。
- 外部排序：指待排序记录的数量很大，以至内存不能容纳全部记录，在排序过程中尚需对外存进行访问的过程。

高级项目经理　任钭
QQ：1530841586



高级项目经理　任钭
QQ：1530841586

向上人生路！

二、插入排序

每步将一个待排序的对象，按其关键码大小，插入到前面已经排好序的一组对象的适当位置上，直到对象全部插入为止。

插入排序有多种具体实现算法：

1) 直接插入排序
2) 希尔排序

高级项目经理　任钭
QQ：1530841586

(1)直接插入排序

先将序列中第1个记录看成一个有序子序列，然后从第2个记录开始，逐个进行插入，直至整个序列有序，排序过程为n-1趟插入。

关键字序列T=（13，6，3，31，9，27，5，11）

【13】, 6, 3, 31, 9, 27, 5, 11

【6, 13】, 3, 31, 9, 27, 5, 11

【3, 6, 13】, 31, 9, 27, 5, 11

【3, 6, 13, 31】, 9, 27, 5, 11

……

【3, 5, 6, 9, 11 , 13 , 27, 31】

高级项目经理　任钭
QQ：1530841586

- 时间效率： 因为在最坏情况下，所有元素的比较次数总和为（0 + 1 + … + n-1)→O(n^2)。故时间复杂度为O(n^2)
- 空间效率：仅占用1个缓冲单元—O(1)
- 算法的稳定性：稳定

高级项目经理　任钬
QQ：1530841586

(2)希尔排序

先取一个正整数d1<n，把所有相隔d1的记录放一组，组内进行直接插入排序；然后取d2<d1，重复上述分组和排序操作；直至di=1，即所有记录放进一个组中排序为止。

一般取d1=n/2, di+1=di/2，如果结果为偶数，则加1。

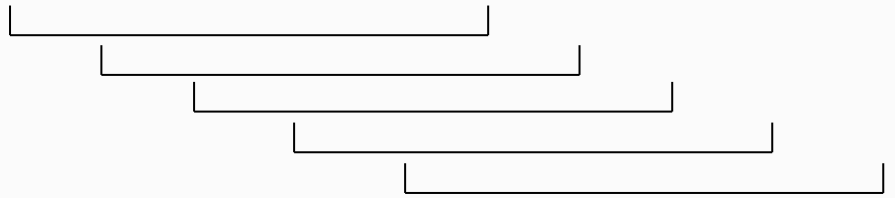
高级项目经理　任钭
QQ：1530841586

高级项目经理　任钭
QQ：1530841586

49 38 65 97 76 13 27 48 55 4

取d1=5

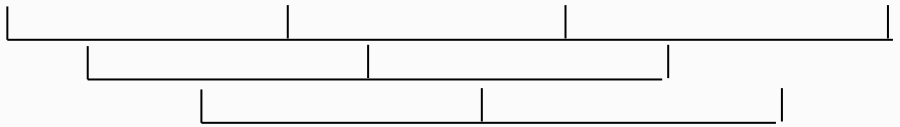
一趟分组：49 38 65 97 76 13 27 48 55 4



一趟排序：13 27 48 55 4 49 38 65 97 76

取d2=3

二趟分组：13 27 48 55 4 49 38 65 97 76



二趟排序：13 4 48 38 27 49 55 65 97 76

取d3=1

三趟分组：13 27 48 55 4 49 38 65 97 76

三趟排序：4 13 27 38 48 49 55 65 76 97

希尔排序是一种不稳定的排序方法

初始序列的不同会影响算法的效率

高级项目经理　任钭

QQ：1530841586

向上人生路！

三、选择排序

每一次从待排序的数据元素中选出最小的一个元素，存放在已排序列的后面，直到全部待排序的数据元素排完。

- 优点：实现简单
- 缺点：每趟只能确定一个元素，表长为n时需要n-1趟
- 前提：顺序存储结构

选择排序分为

(1)直接选择排序

(2)堆排序

高级项目经理　任钭
QQ：1530841586

(1)直接选择排序

在所有记录中选出最小的记录，把它与第1个记录交换，然后在剩余的记录内选出最小的记录与第2个交换.....依次类推。

例：关键字序列T=（21，25，49，27，16，08）

第1趟 08，25，49，27，16，21

第2趟 08，16，49，27，25，21

第3趟 08，16，21，27，25，49

第4趟 08，16，21，25，27，49

第5趟 08，16，21，25，27，49

高级项目经理 任铄

QQ：1530841586

高级项目经理　任钬
QQ：1530841586

(2)堆排序

堆是满足下列性质的数列{r1, r2, … , rn}

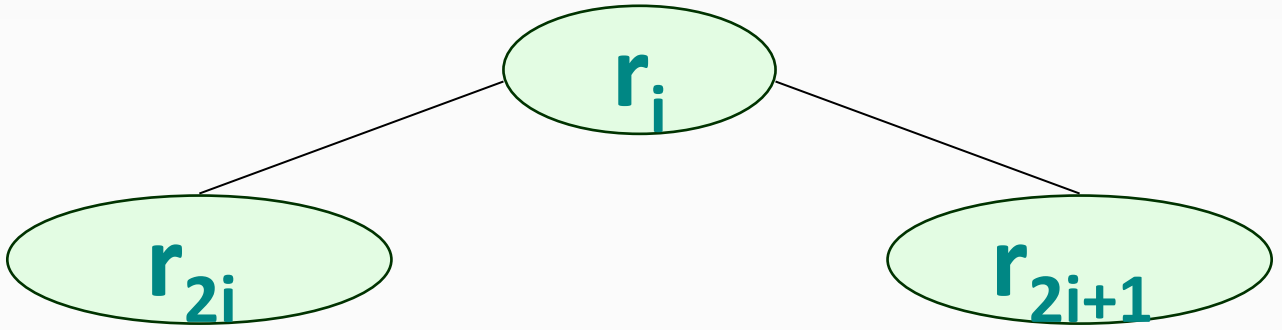
$$\begin{cases} r_i \le r_{2i} \\ r_i \le r_{2i+1} \end{cases}$$ (小顶堆)　或　$$\begin{cases} r_i \ge r_{2i} \\ r_i \ge r_{2i+1} \end{cases}$$ (大顶堆)

例如：

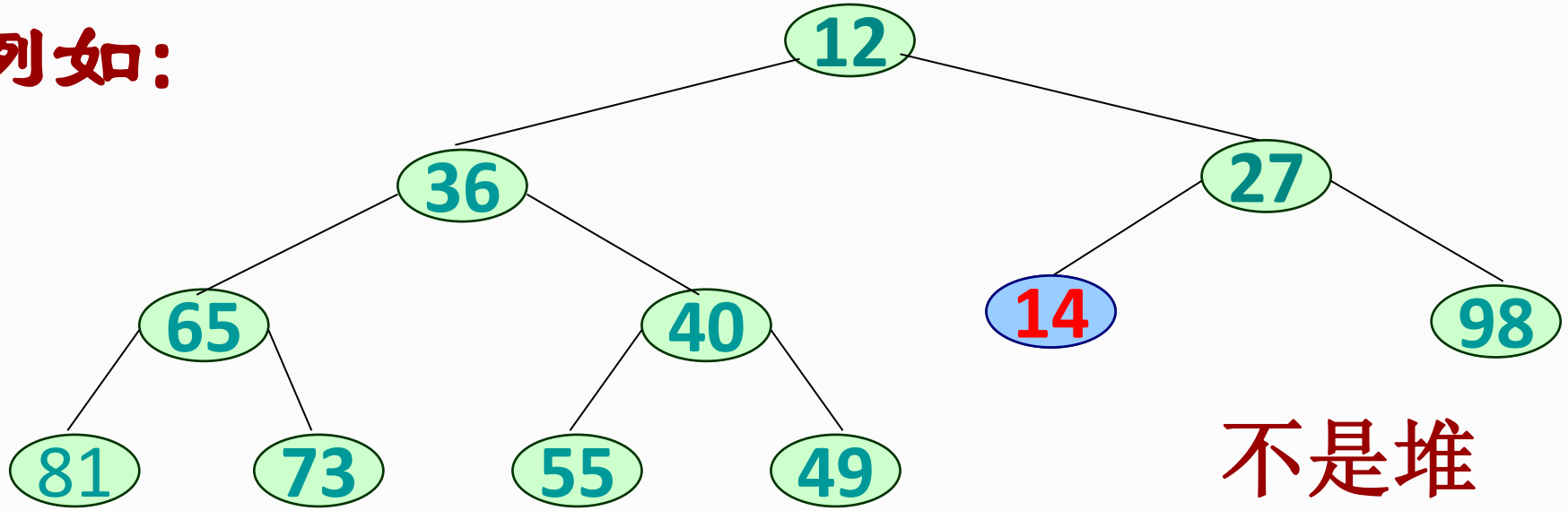
{12, 36, 27, 65, 40, 34, 98, 81, 73, 55, 49}　是小顶堆

{12, 36, 27, 65, 40, 14, 98, 81, 73, 55, 49}　不是堆

若将该数列视作完全二叉树，则 r_{2i} 是 r_i 的左孩子； r_{2i+1} 是 r_i 的右孩子

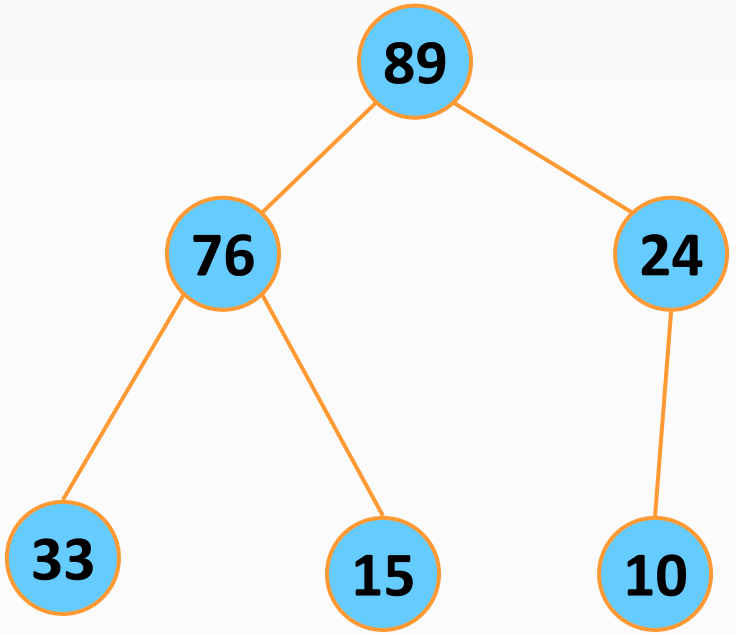


例如：

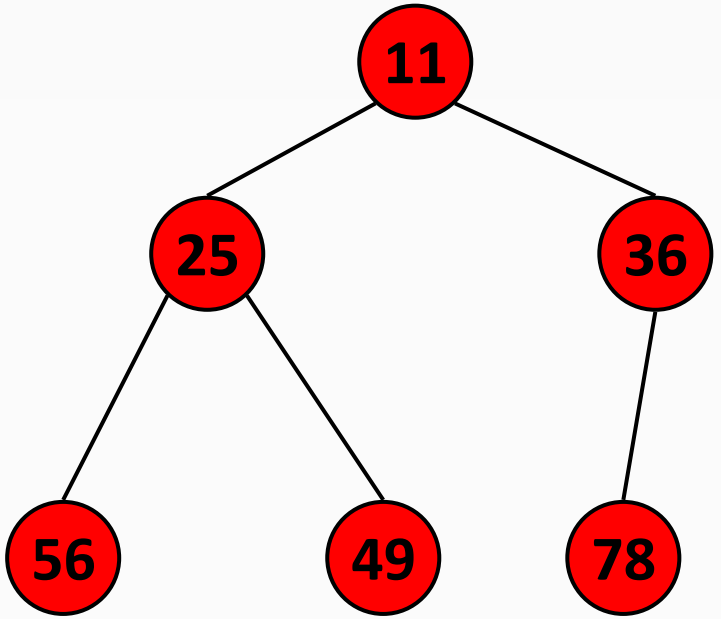


不是堆

向上人生路！



大顶堆



小顶堆

高级项目经理　任钭
QQ：1530841586

向上人生路！

堆排序即是利用堆的特性对记录序列进行排序的一种排序方法。

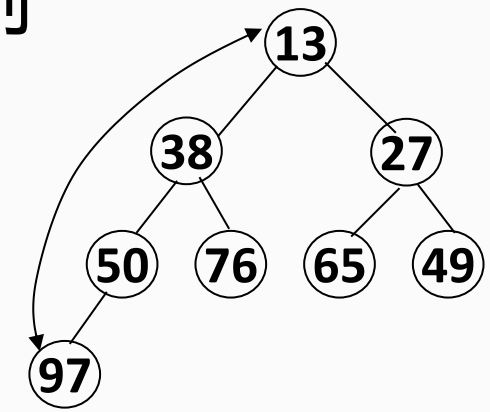
将无序序列建成一个堆，得到关键字最小（或最大）的记录；输出堆顶的最小（大）值后，使剩余的n-1个元素又建成一个堆，则可得到n个元素的次小值；重复执行，得到一个有序序列，这个过程叫堆排序。

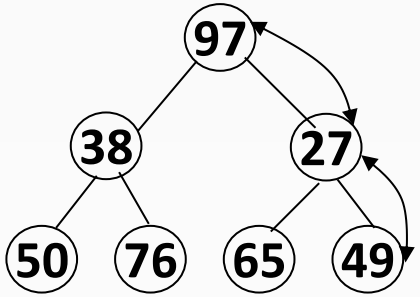
高级项目经理　任铄
QQ：1530841586

输出堆顶元素之后，以堆中最后一个元素替代之；然后将根结点值与左、右子树的根结点值进行比较，并与其中小者进行交换；重复上述操作，直至叶子结点，将得到新的堆，称这个从堆顶至叶子的调整过程为“筛选”。

高级项目经理　任钭
QQ：1530841586

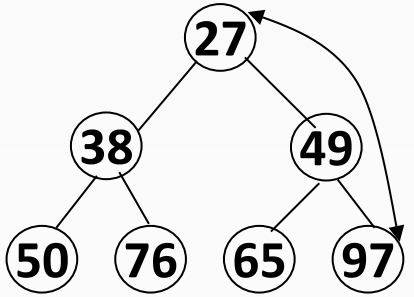
例





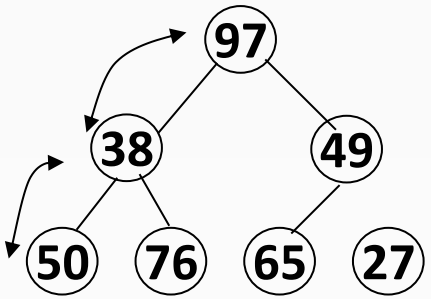
13

输出：13



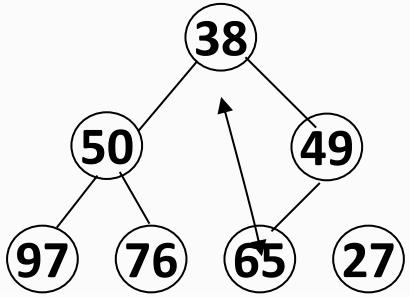
13

输出：13



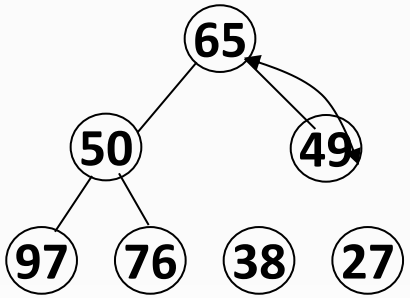
13

输出：13 27



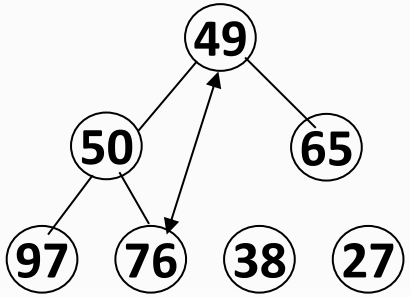
13

输出：13 27



13

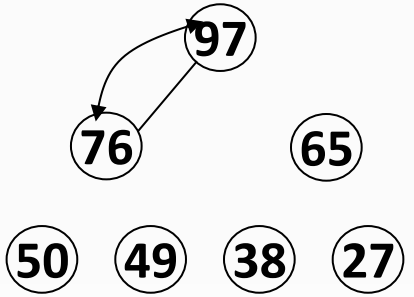
输出：13 27 38



13

输出：13 27 38

……



13

输出：13 27 38 49 50 65

堆排序的最坏时间复杂度为$O(nlog_2n)$，堆排序的平均性能接近最坏性能。

堆排序的辅助空间为O(1)。

高级项目经理　任钬
QQ：1530841586

可以通过下列渠道沟通联系：

1、 QQ:1530841586

2、 QQ群：164955673

向上人生路！