1. 数据结构中（逻辑结构）分为四类基本的结构：
2. 集合结构，结构的数据元素关系是属于“同一个集合”（无逻辑关系）
3. 线性结构，结构的数据元素之间存在一对一的关系。（常用的线性结构，线性表，栈，队列，双队列，数组，串）

特点：

1.有唯一的首元素（第一个元素），

2.有唯一的尾元素（最后一个元素），

3.除首元素外，所有元素有且只有唯一的前驱，

4.除尾元素外，所有元素有且只有唯一的后续，

1. 树形结构，结构的数据元素之间存在着一对多的关系。（树）
2. 图形结构，结构的数据元素之间存在着多对多的关系，也称网状结构。图形结构中，每个结点的前驱结点数和后续结点数可以任意多个。数据关系是任意的。图形结构中任意两个数据元素间均可相关联。

非线性结构的特点与线性结构的特点相反。

**Tips：树形结构和图形结构（网状结构）统称为非线性结构。**

1. 数据结构的三要素：逻辑结构，存储结构，数据运算。
2. 数据结构中（存储结构（物理结构））表述的是含有某种逻辑关系的元素在计算机中存储的方式。可以理解为数据元素在存储器上的排列方式。

分为四种：顺序存储，链式存储，索引存储，散列存储（哈希存储）

顺序存储，将逻辑上相邻的数据元素，存储到计算机的存储器上，在计算机存储器中也是相邻的，即内存之间是相邻的，连续的，例如数组，优点在于可以随机存取，即通过数组下标我们可以很轻松找到数据元素或者修改它。

链式存储：例如链表，用指针将可能不在连续内存地方的内容连接起来形成一个表。

缺点很明显，想找到编号n的元素，只能从表头开始遍历。

索引存储：书和目录的关系，找到对应的只需查看目录，一般存储的时候{关键字，地址}这种形式，类似python中的字典。

散列存储：实际就是一个函数关系的映射。

算法时间复杂度：(一般说的算法时间复杂度是平均时间复杂度)

3个常用的结论：

1. 加数常数项可以省略：  
   eg：A(2n+3) B(5n+1) 可以写成A(2n) B(5n)
2. 除去最高阶项，其他次项可以省略

eg: A(2n^2 + 2n+2) B(2n^2) lim n->∞A = B

1. 与最高次项相乘常数可省

Eg: A(2n^2) ->(n^2) B(10n) ->(n)

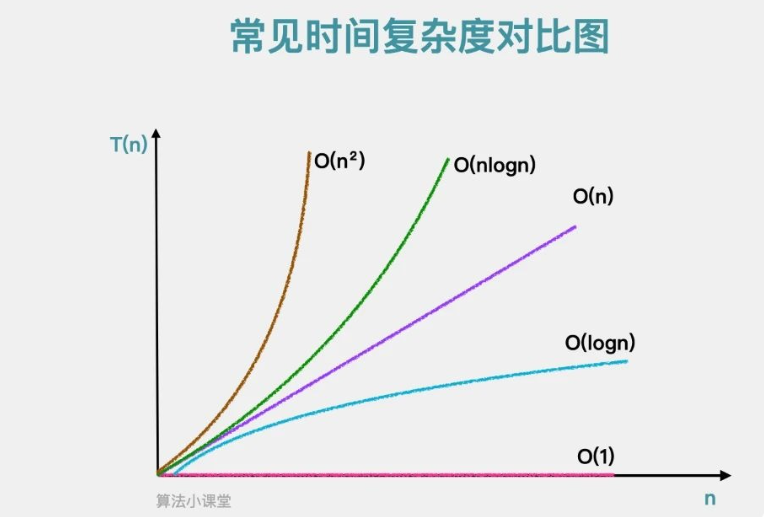
常数项在大O记法，用常数1取代运行时间中的所有加法常数。T(n) = O(f(n)) (f(n)为运行的次数，T(n)为时间复杂度)，eg:f(n) = 3, T(n) = O(3) = O(1)

对数阶O(logn)

线性阶O(n)

线性对数阶 O(nlogn)

平方阶 O(n^2)



表达方式：

O（f(n)）表达最差性能，表示小于等于的意思

Ω（g（n））表达最好的时间，即算法的下界，表示大于等于的意思

Θ需同时满足大Ο和Ω，故称为确界（必须同时符合上界和下界），表示等于的意思

o表达上界，小于的意思

ω表达下界，大于的意思

空间复杂度 S(n)=O(f(n)) (n为问题的规模，f(n)为语句关于n所占存储空间的函数)

1.表：

创建基本的表分为几步：

第一步：对表进行初始化，即对结构体里的元素进行初始化;

第二步：测试表是否为空;

第三步：设计计算表长度功能。

第四步：设计表L的位置k的元素

第五步：设计表L中元素x的索引

第六步：设计向表L的位置K之后插入元素x

第七步：设计删除表L位置K处的元素

第八步：按位置次序输出表L中的元素

2.栈：

栈的特点：后进先出（特殊的表）

可以用数组和指针来实现；

创建基本的栈分为几步：

定义好结构体：

1. 对栈进行初始化
2. 查看栈顶是否为空
3. 返回栈顶结点的元素
4. 插入元素
5. 删除元素

3.队列：

队列特点：先进先出（特殊的表）

可以用链表实现队列，也可用循环数组进行实现；

创建基本的队列分为记不：

定义好结构体：

1. 对队列进行初始化
2. 查看队列是否为空
3. 返回队首和队尾的元素
4. 向队尾插入数据
5. 对队首的元素进行删除
6. 排序与选择算法
7. 简单排序法 O(n^2)
8. 冒泡排序法：两两元素进行比较，大的往上冒，下的往下排。
9. 插入排序法

将第一个元素标记为已排序

对于每一个未排序的元素 X

  “提取” 元素 X

  i = 最后排序过元素的索引 到 0 的遍历

    如果当前元素 j > X

      将排序过的元素向右移一格

    跳出循环并在此插入 X

1. 选择排序法

重复（元素个数-1）次

  把第一个没有排序过的元素设置为最小值

  遍历每个没有排序过的元素

   如果元素 < 现在的最小值

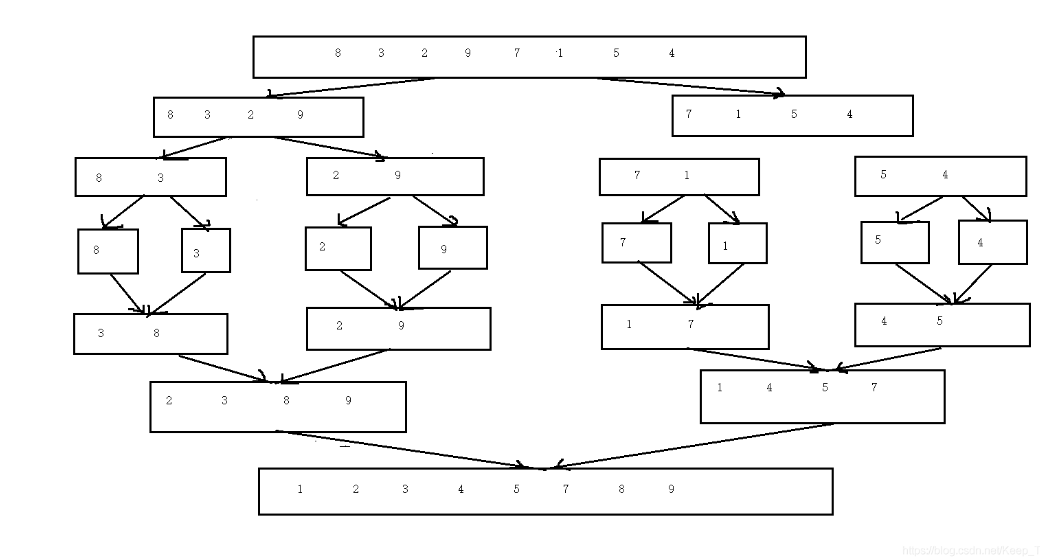
      将此元素设置成为新的最小值

  将最小值和第一个没有排序过的位置交换

1. 快速排序算法O(nlogn)
2. 快速排序算法：
   1. 以一个值为基准值，以i，j分别从左向右，从右向左，然后将小于基准值的放于基准值左边，大于的放于基准值右边。如果i==j的话就停止寻找，将基准值与a[j]交换，然后递归循坏
3. 随机快速排序法：
   1. 在快速排序算法上，把基准值随机的在数组a中取。然后执行快速排序算法。

Tips：以上的快速排序算法以及随机快速排序算法，因为运用递归，所以对计算机的堆栈，消耗计算机资源，所以有了非递归快速排序算法

1. 即模拟递归的方式进行快速排序：
   1. 用自己之前的实现的栈进行模拟函数的调用
2. 对快速排序算法的优化可以在选择基准值的时候进行优化。
3. 合并排序算法O(nlogn) 渐近最优算法
4. 合并排序是采用分治策略实现对N个元素进行排序的算法，是分治法的一个典型应用和完美体现，它是一种平衡，简单的二分分治策略，计算过程分为三步：
5. （1）分解：将待排序元素分成大小大致相同的两个子序列。
6. （2）求解子问题：用合并排序法分别对两个子序列递归地进行排序。
7. （3）合并：将排好序的有序子序列进行合并，得到符合要求的有序序列。



后面链表合并算法，自顶向下合并，自然合并算法都是处理分为无法再分的数组的时候进行合并优化的算法。

1. 计数排序算法
   1. 对原先的数组进行每个元素进行计数，然后将计数的写入对应的新数组。之后将元素一一写入原数组。
2. 桶排序算法
3. C语言新知识点：函数作为一个参数传递给另一个参数，在函数中进行函数调用。
   1. Eg：void preorder(函数形参) 函数形参里需要定义一个函数指针意味着这个参数是指向函数的第一块内存。函数形参 void(\*visit)(int a,int b) 类似这样的意思为\*visit为一个函数指针指向一个函数，指向的函数要满足返回值是void并且指向的函数的形参是(int ,int)类型的。所以被指向的函数void functionname (int ,int)这样的。

算法与数据结构基础知识：

1. 算法的基本特征：可行性，确定性，有穷性，输入和输出
2. 二叉树度为0的结点总是比度为2的结点多一个。