算法的5个重要特性：有穷性，确定性，可行性，输入，输出

时间复杂度的计算：

线性表：数组实现，链表实现(单向/双向)

栈：先进后出的线性表(所以，栈可以通过数组或链表实现)

栈的使用：

10进制与其他进制之间的转换

括号匹配 [ ( [ ] [ ] ) ]问题

迷宫问题：有一个入口，一个出口，将迷宫中的分支进行压栈，当某条路不能再走时，将之间的路径进行弹栈，回到最近的分支处，重新压栈

4+2\*3-10/5的运算顺序，符号放一个栈中，数字放一个栈中

递归的实现就是通过栈实现的

队列：先进先出的线性表(循环队列)

字符串匹配(KMP算法)：

矩阵的压缩存储：

如果值相同的元素或零元素在矩阵中分布有一定的规律，此类矩阵为特殊矩阵，反正为稀疏矩阵

对称矩阵可以使用数组实现，数组存放的元素为对称矩阵的一个三角矩阵，长度为n(n+1)/2

数组下标k和矩阵下标i，j之间的关系：

i>=j时：k=i(i-1)/2+j-1

i<j时 ：k=j(j-1)/2+i-1

三角矩阵，对角矩阵都可以使用数组的方式进行保存，压缩

对于稀疏矩阵，零元素很多，非零元素很少，且无规律，在保存的时候，可以保存非零元素的i，j及其对应的值

树：

树的子数个数称为结点的度

树的深度：有多少层

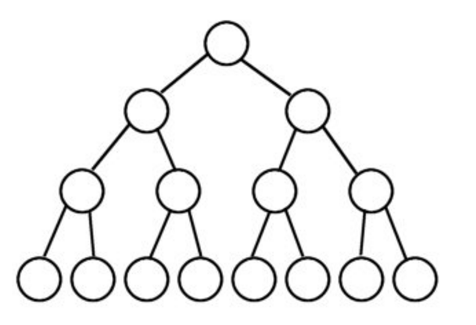
二叉树的性质：

二叉树的第i层，最多有2^(i-1)个节点

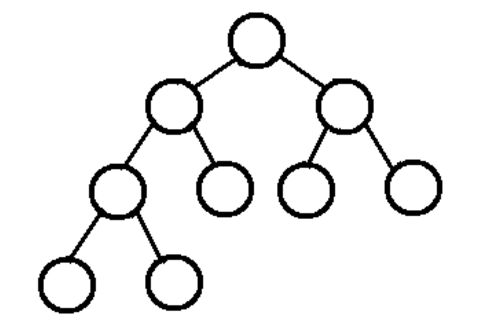
深度为k的二叉树最多有2^k-1个节点

完全二叉树/满二叉树

满二叉树：深度为k，k层节点数为2^k-1



完全二叉树：在满二叉树的基础上，从最底层，从右开始删除节点



二叉树的存储：数组和链式

二叉树的遍历：

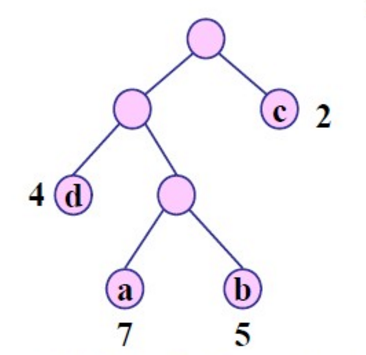
先序：根，左，右

中序：左，根，右

后序：左，右，根

通过前序找到根节点，中序找到根节点的左右支

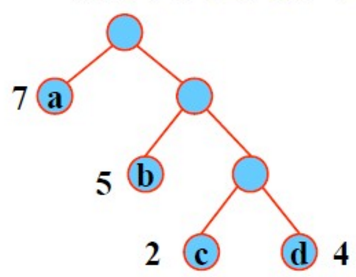
最优二叉树(哈夫曼树)：



每个叶子节点都带权，带权路径长度是指到达节点所需的时间

A：7\*3=21，B：5\*3=15，C：2\*1=2，D：4\*2=8 权值\*深度

在查询A的时候，带权路径就会比较长，应将权值大的存放到深度浅的



这就是一个哈夫曼树



所以在判断时候应该先从80开始判断，先判断概率(权值)大的

