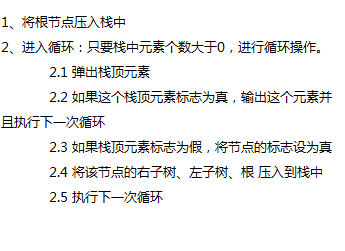
1. 队列---顺序存储
   1. 利用写好的动态数组实现 顺序存储的队列数据结构
   2. 对外提供的接口
   3. 初始化队列
   4. 入队
   5. 出队
   6. 返回队头
   7. 返回队尾
   8. 返回队列大小
   9. 销毁队列
   10. 判断队列是否为空
2. 队列---链式存储
   1. 和顺序存储对外接口一致
   2. 利用链表方式实现
   3. 实现出一种 符合 先进先出的数据结构
3. 树的基本概念
   1. 根节点 没有前驱的节点
   2. 叶子节点 没有后继节点
   3. 双亲 前驱节点
   4. 孩子 后继节点
   5. 节点的度 该节点的直接后继的数量
   6. 树的度 节点的度中 最大的值
   7. 树的高度 树的层数
4. 二叉树性质
   1. 性质1: 在二叉树的第i层上至多有2i-1个结点（i>0）
   2. 性质2: 深度为k的二叉树至多有2k-1个结点（k>0）
   3. 性质3: 对于任何一棵二叉树，若度为2的结点数有n2个，则叶子数（n0）必定为n2＋1 （即n0=n2+1）
   4. 满二叉树 结点数量为 2k-1
   5. 完全二叉树 除了最后一层结点，上面是一颗满二叉树，最后一层的节点尽量往左靠
   6. 性质4: 具有n个结点的完全二叉树的深度必为⎣log2n⎦+1
   7. (如 log2 (15)    点击 15  log  /  2  log =)
   8. 性质5: 对完全二叉树，若从上至下、从左至右编号，则编号为i 的结点，其左孩子编号必为2i，其右孩子编号必为2i＋1；其双亲的编号必为i/2（i＝1 时为根,除外）
5. 二叉树的递归遍历
   1. 利用树递归特性可以将二叉树进行递归遍历
   2. 先序遍历 先根 再左 再右
   3. 中序遍历 先左 再根 再右
   4. 后序遍历 先左 再右 再根
   5. 利用代码实现二叉树递归遍历
6. 二叉树编程
   1. 利用递归特性 求出二叉树中的叶子的数量
   2. 求出二叉树的高度
   3. 利用递归特性 拷贝出二叉树
7. 二叉树的非递归遍历
   1. 利用栈 实现
   2. 首先将每个节点都设置一个标志，默认标志为假，根据节点的的状态进行如下流程
   3. 
8. 插入排序
   1. 外层大循环 从下标为1开始
   2. 内存小循环 ，满足一定条件后，数据后移
   3. 将数据插入到相应的位置上 j + 1位置