1、查找树：用于处理动态查找表的树

注：一棵结构良好的二叉树可以以O(logN)支持插入和删除；如果这棵树有序，可以做到O(logN)支持查找

2、二叉查找树（二叉排序树）

（1）定义：或者是一棵空树，或者满足

1°若左子树不空，则左子树中的所有元素的键值都比根结点的键值小

2°若右子树不空，则右子树中断 所有元素的键值都比根结点的键值大

3°左、右子树均为二叉查找树

—>重要性质：中序遍历一棵二叉查找树，得到的序列按键值递增

（2）基本操作

1°find(x)：查找

2°insert(x)：插入

3°remove(x)：删除

3、AVL树（平衡树、二叉平衡查找树）

（1）左右子树的高度差在一定的范围内，使得二叉查找树尽可能矮胖，但不一定是完全二叉树（为了解决二叉查找树退化为单链表后树太高的问题）

（2）结点的平衡度（平衡因子）：结点的左子树的高度减去右子树的高度

注：AVL树的每个结点的平衡度只能为-1,0,1

4、AVL树的插入操作：插入+恢复平衡

（1）造成结点A失衡的四种情况

1°LL情况

2°RR情况

3°LR情况

4°RL情况

（2）解决LL和RR情况的办法——单旋转

LL情况：向右旋转

RR情况：向左旋转

（3）解决LR和RL情况的办法——双旋转

LR情况：先左单旋转，再右单旋转

RL情况：先右单旋转，再左单旋转

5、AVL树的删除操作：删除+恢复平衡