### AVL树功能实现原理实现

1、平衡二叉树插入功能实现(opadd = true)

步骤分析：

1）判断当前二叉树是否平衡，不平衡提醒进行调整处理

2）平衡时，若root 为空，则 插入到根节点

3）若root 不为空，比较插入值跟根节点大小

1、若插入值等于根节点，则返回-1，说明该节点已经存在

2、若插入值小于根节点，则与根节点的左孩子比较，循环（3），直到节点的左孩子为空，插入到该节点的左节点 。

3、若插入值大于根节点，则与根节点的右孩子比较 ，循环（3），直到节点的右孩子为空，插入到该节点的右节点 。

4）更新插入点到根接点的路径上的所有节点的高度（hight）和平衡因子(balance)

若存在平衡因子为-2 或2 的节点时 ，把该节点赋值给A;

二叉树树的ava置为false;

2、平衡二叉树查找功能实现

若root 不为空，比较插入值跟根节点大小，

1、若插入值等于根节点，则返回根节点

2、若插入值小于根节点，则与根节点的左孩子比较，

3、若插入值大于根节点，则与根节点的右孩子比较 ，

4、循环（1-3），直到节点的右孩子或左孩子为空，返回NULL,说明该节点不存在。

3、平衡二叉树删除功能实现(opadd = false)

步骤分析：

1)判断当前二叉树是否平衡，不平衡提醒进行调整处理

2)平衡时，调用查找功能，找到删除的节点。若节点为空、返回删除的节点不存在

3)若节点不为空，则 查找左子树的最大值，即中序遍历的前驱替代、递归删除，直到删除叶子节点停止。

4)更新删除后的叶子接点到根接点的路径上的所有节点的高度和平衡因子，若存在平衡因子为-2 或2的节点时，把该节点赋值给A，并把二叉树树的ava置为false;

4、平衡二叉树平衡功能实现

可以通过旋转根及其孩子的BF值来决定作什么类型的旋转操作：

Insert 时

1） 当旋转根的BF值为2时：

如果旋转根的左孩子的BF值为1，则进行LL型旋转；

如果旋转根的左孩子的BF值为-1，则进行LR型旋转。

2)  当旋转根的BF值为-2时：

如果旋转根的右孩子的BF值为1，则进行RL型旋转；

如果旋转根的右孩子的BF值为-1，则进行RR型旋转。

Delete 时

1） 当旋转根的BF值为2时：

如果旋转根的左孩子的BF值为大于 -1，则进行LL型旋转；

如果旋转根的左孩子的BF值为-1，则进行LR型旋转。

2)  当旋转根的BF值为-2时：

如果旋转根的右孩子的BF值为1，则进行RL型旋转；

如果旋转根的右孩子的BF值为小于1，则进行RR型旋转。