# 平衡二叉查找树

1. 平衡二叉树的性质
   1. 平衡二叉树是二叉查找树的一种，且具有以下性质：它是一颗空树或者它的任意节点的两个子树的高度差不超过1，并且左右两个子树都是平衡二叉树。这个方案很好的解决了二叉查找树退化成链表的问题。
2. 平衡二叉树的插入
   1. 插入导致二叉树不平衡的情况

把不平衡的树或子树的根节点叫做a，如果某一次插入导致a的左右子树高度差超过1，有以下四种情况

* + 1. 对α的左儿子的左子树进行一次插入
    2. 对α的左儿子的右子树进行一次插入
    3. 对α的右儿子的左子树进行一次插入
    4. 对α的右儿子的右子树进行一次插入
  1. 调整措施
     1. 对于左左情况只需对树进行一次旋转即可，把a的左儿子调整为根节点即可，右右情况则是左左情况的镜像。
     2. 对于左右情况则需进行两次旋转，第一次旋转针对a的左子儿子k1，将k1下降一个层级，而将k1的右儿子上升一个层级，将左右情况调整为左左情况。接下来再进行一次左左旋转即可。

1. 平衡二叉树的删除

删除分为以下几种情况

* 1. 要删除的是当前根节点T
     1. 如果左右子树中有一个为空，那么直接用那个非空子树或者是NULL替换当前根节点即可。
     2. 左子树高度大于右子树高度，将左子树中最大的那个元素赋给当前根节点，然后删除左子树中元素值最大的那个节点。
     3. 左子树高度小于右子树高度，将右子树中最小的那个元素赋给当前根节点，然后删除右子树中元素值最小的那个节点。
  2. 要删除的节点元素值小于当前根节点T值，在左子树中进行删除。

递归调用，把要删除的节点当做根节点，实施删除。删除后需要判断当前根节点是否任满足平衡条件，如果满足平衡条件，只需要更新当前根节点T的高度信息。否则，需要进行旋转调整：

如果T的左子节点的左子树的高度大于T的左子节点的右子树的高度，进行相应的单旋转。否则进行双旋转。

1. 要删除的节点元素值大于当前根节点T值，在右子树中进行删除。