Lesson09--04--搜索结构之AVL树

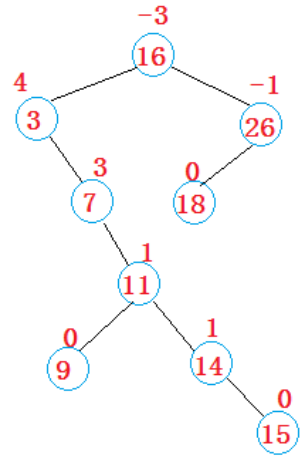
【本节目标】
1、AVL树概念
2、AVL实现原理
2、平衡化旋转：单旋转(左单旋&右单旋)和双旋转(先左后右双旋&先右后左双旋)
3、AVL树插入

二叉搜索树虽可以缩短查找的效率，但如果数据有序或接近有序二叉搜索树将退化为单支树，查找元素相当于在顺序表中搜索元素，效率低下。

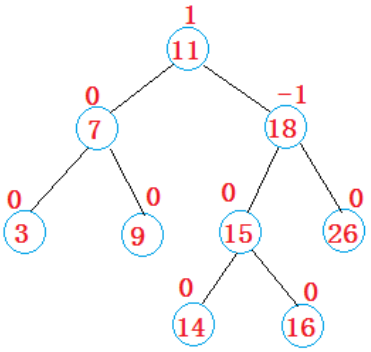
因此，两位俄罗斯的数学家G.M.Adelson-Velskii和E.M.Landis在1962年发明了一种解决上述问题的方法：当向二叉搜索树中插入新结点后，如果能保证每个结点的左右子树高度之差的绝对值不超过1(需要对树中的结点进行调整)，即可降低树的高度，从而减少平均搜索长度。

- AVL树概念

一棵AVL树或者是空树，或者是具有以下性质的二叉搜索树：
1、它的左右子树都是AVL树
2、左子树和右子树高度之差(简称平衡因子)的绝对值不超过1(-1、0、1)



高度不平衡的搜索二叉树



高度平衡的搜索二叉树

如果一棵二叉搜索树是高度平衡的，它就是AVL树。如果它有n个结点，其高度可保持在O(lgn)，平均搜索时间复杂度O(lg(n))

- AVL树的实现原理

假设有一个数组a[10] = {3,2,1,4,5,6,7,10,9,8},在未接触平衡二叉树之前，根据二叉搜索树的特性，构建如图左所示的样子

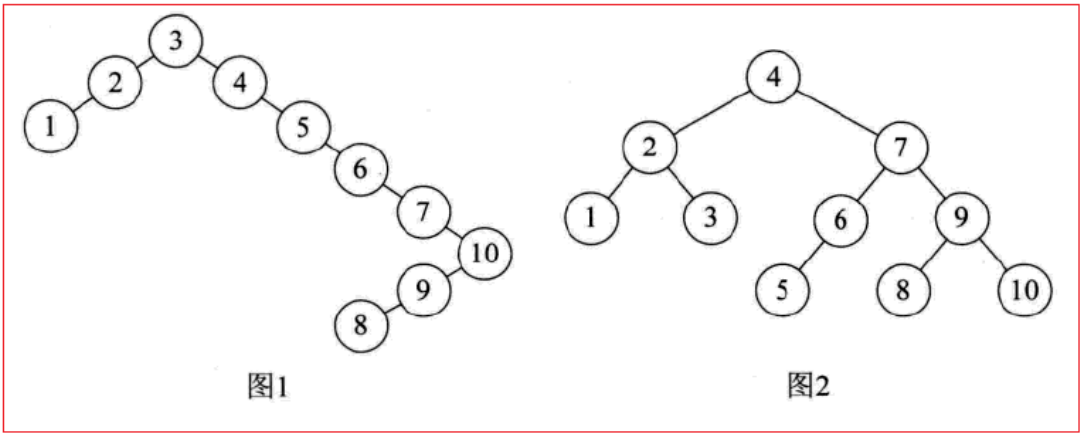
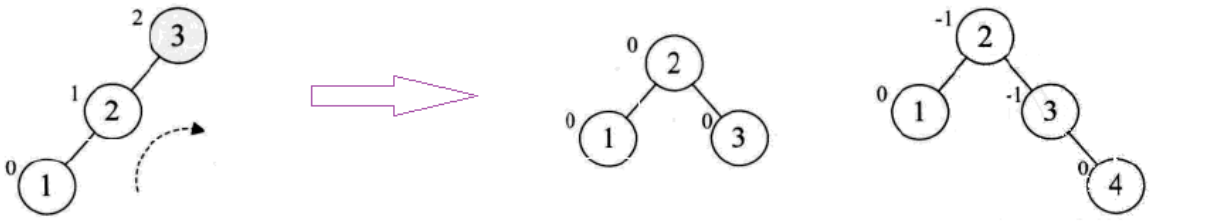
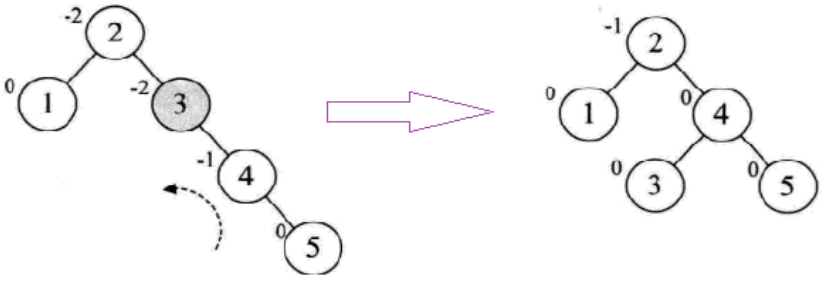


图2树的构架过程

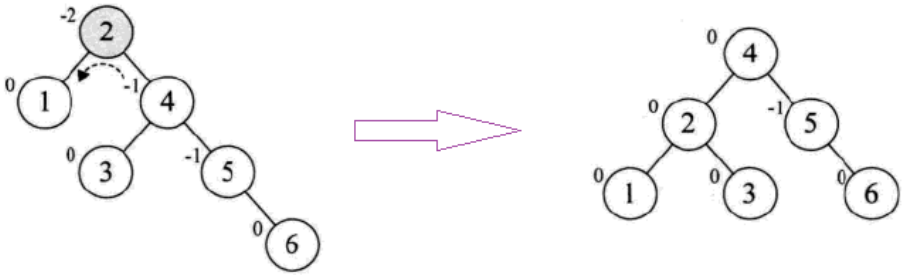


插入1后3不平衡，需要调整树

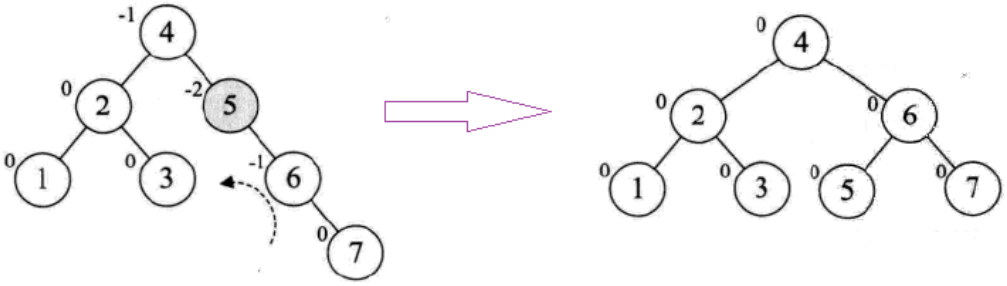
插入4后个节点仍旧保持平衡



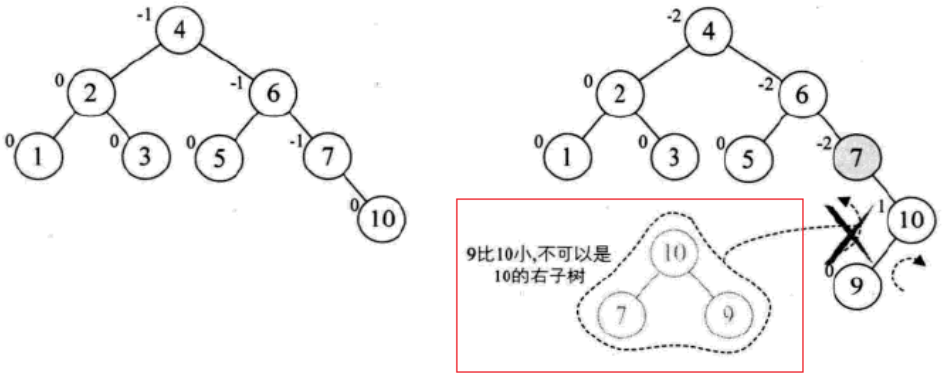
插入5后3不平衡，需要调整树



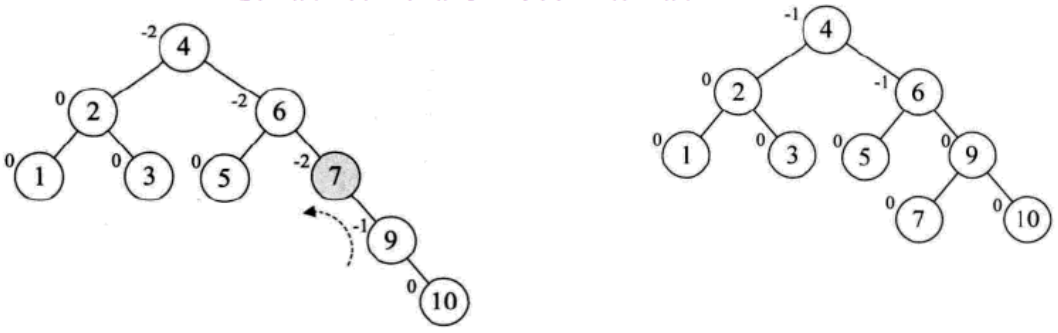
插入6后2不平衡，需要调整树



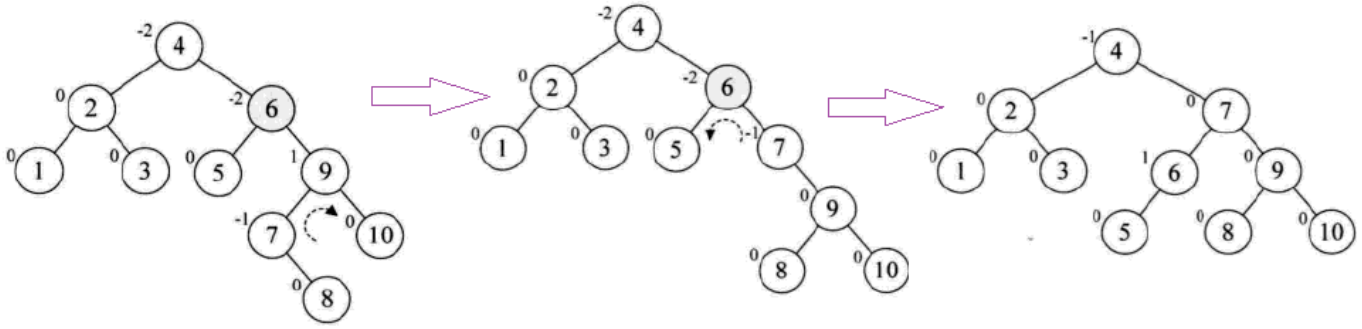
插入7后5不平衡需要调整



插入9后，7、6、4均不平衡，都有可能需要调整，
先让7保证平衡，如果直接以10为轴左旋转，旋转之
后虽然平衡但满足搜索二叉树，所以不能以10为轴
直接左旋，那么只能以10为轴进行右旋



该种情况前面已经出现过，直接以7为轴，进行左旋，再
次平衡

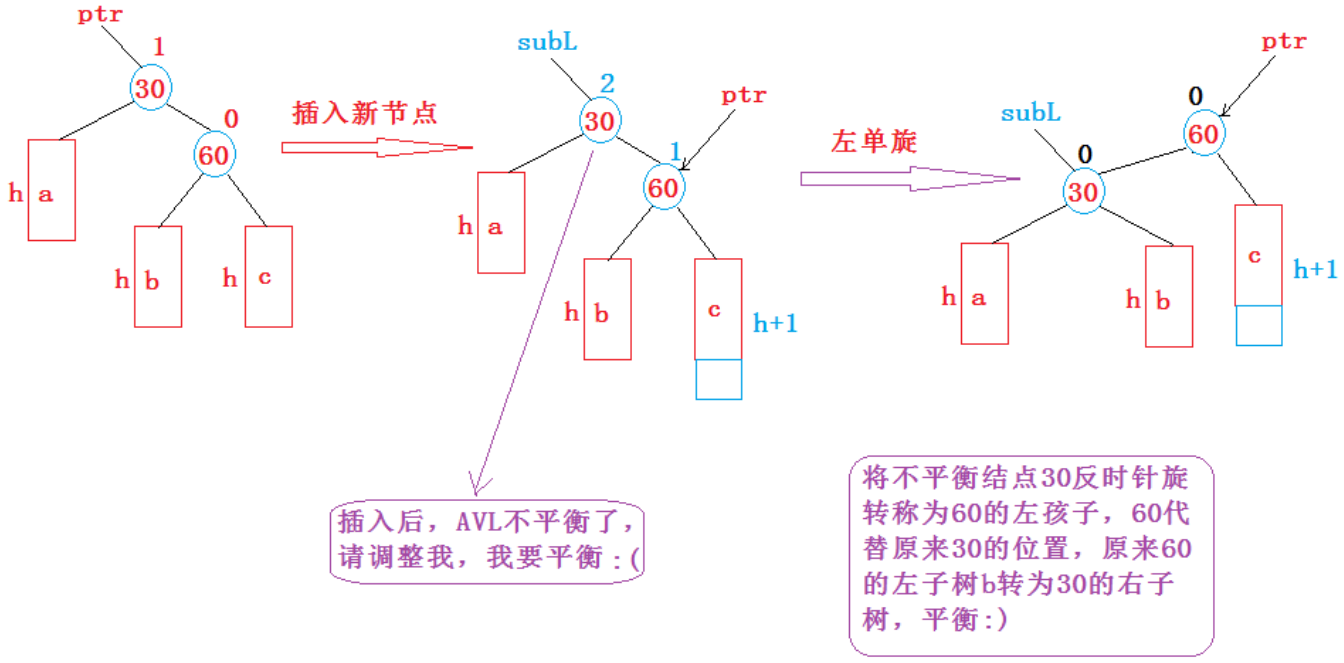


插入8后6、2不平衡，先将9右旋，然后再将7左旋

- 平衡化旋转

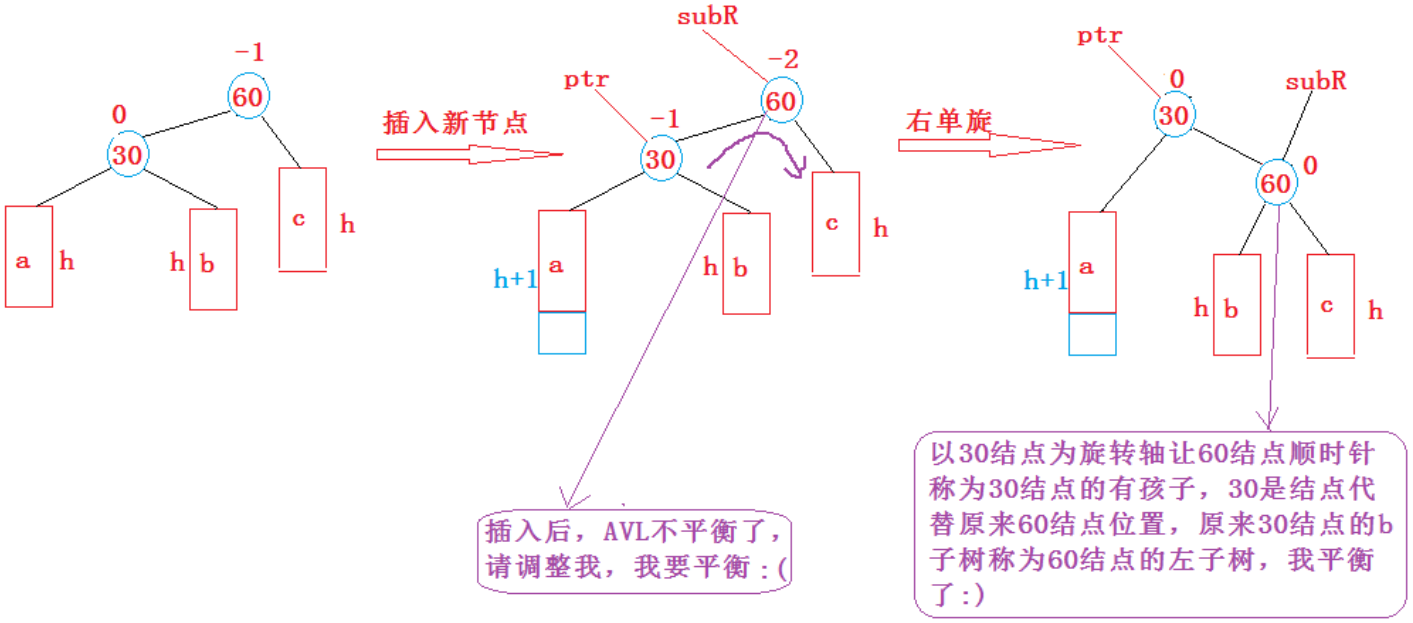
如果在一棵原本是平衡的二叉搜索树中插入一个新节点，可能造成不平衡，此时必须调整树的结构，使之平衡化。

【左单旋】
插入点位于ptr的右子节点的右子树--右右



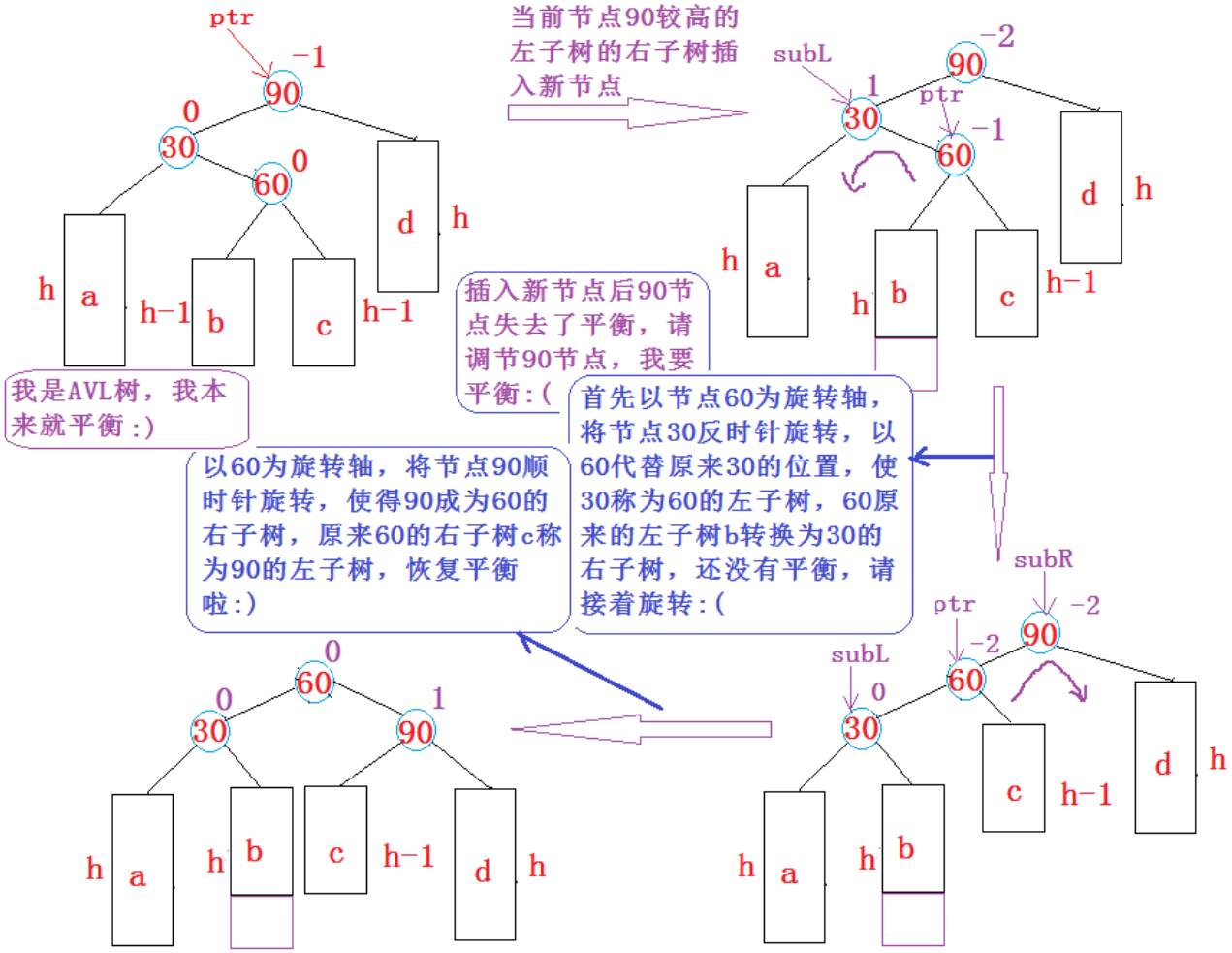
【右单旋】

插入点位于ptr的坐姿结点的左子树--左左



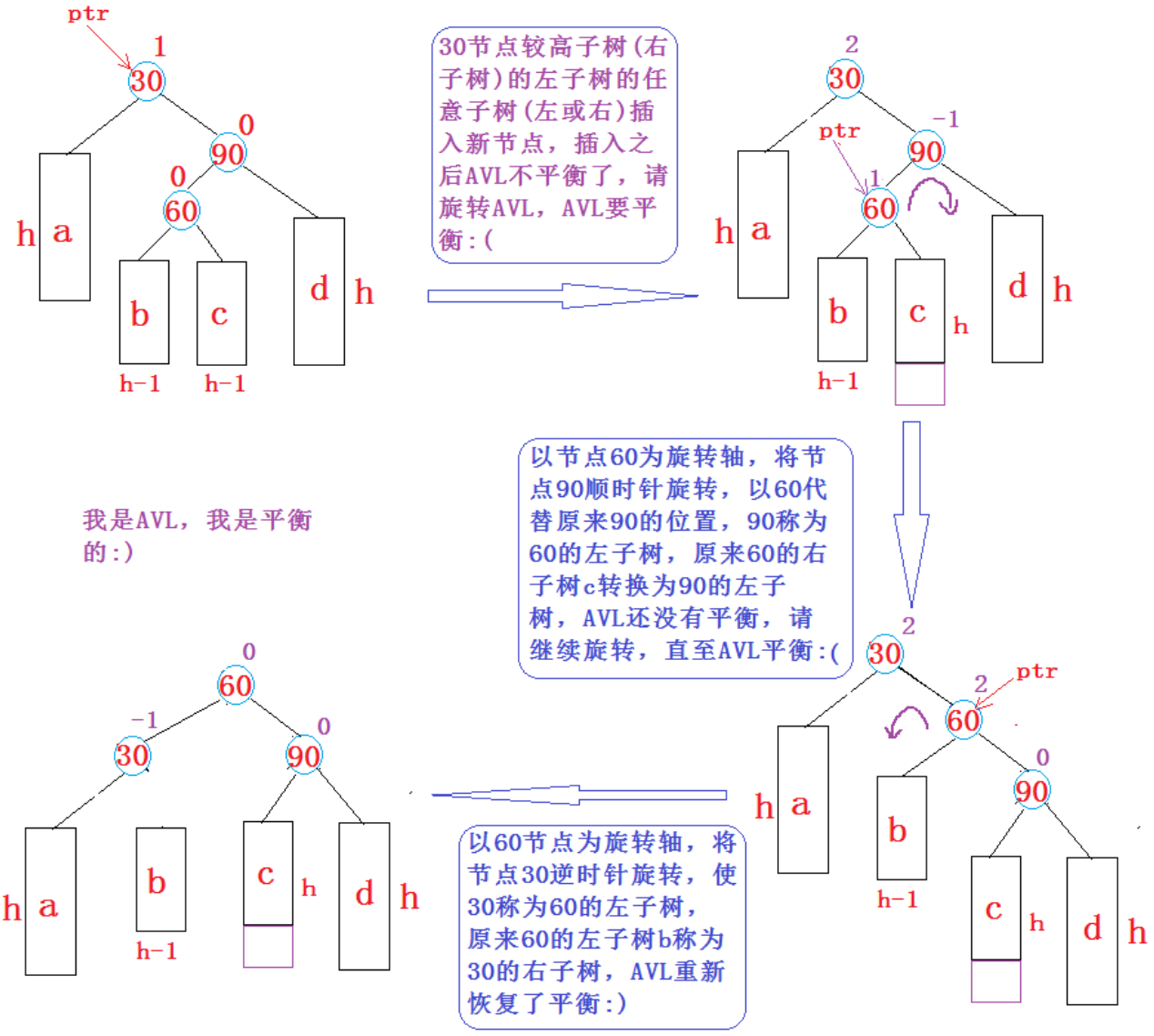
【先左后右双旋】

插入点位于ptr左子节点的右子树--左右



【先右后左双旋转】

插入点位于ptr右子节点的左子树--右左



- AVL树的插入

在向一棵本来高度平衡的AVL树中插入一个新节点时，如果树中某个结点的平衡因子的绝对值>1，则出现了不平衡。设新插入结点为P，从结点P到根节点的路径上，每个结点为根的子树的高度都可能增加1，因此在每执行一次二叉搜索树的插入运算后，都需从新插入的结点P开始，沿该结点插入的路径向根节点方向回溯，修改各结点的平衡因子，调整整棵树的高度，恢复被破坏的平衡性质。

在AVL树中插入结点P(key)结点算法：
步骤一：如果是空树，插入后即为根节点，插入后直接返回
步骤二：如果树不空，寻找插入位置，若在寻找的过程中找到key，则插入失败直接返回
步骤三：插入结点
步骤四：更新平衡因子，对树进行调整

新节点P的平衡因子为0，但其双亲结点Pr的平衡因子有三种情况：
1、结点Pr的平衡因子为0
在Pr的较矮的子树上插入新节点，结点Pr平衡，其高度没有增加，此时从Pr到根路径上各结点为根的子树的高度不变，即各结点的平衡因子不变，结束平衡化处理。

2、结点Pr的平衡因子的绝对值为1；
插入前Pr的平衡因子为0，插入后以Pr为根的子树没有失去平衡，但该子树的高度增

加，需从该结点Pr向根节点方向回溯，继续查看Pr的双亲结点的平衡性。

3、结点Pr的平衡因子的绝对值为2

新节点在较高的子树插入，需要做平衡化处理：

若Pr = 2，说明右子树高，设Pr的右子树为q

当q的平衡因子为1，执行左单旋转

当q的平衡因子为-1，执行先右后左双旋转

若Pr = -2，说明左子树高，设Pr的左子树为q

当q的平衡因子为-1，执行右单旋转

当q的平衡因子为1，执行先左后右双旋转

旋转后Pr为根的子树高度降低，无需继续向上层回溯

- AVL树的删除

从AVL树中删除一个节点，首先必须检测该结点是否存在，若存在删除该结点之后可能会破坏AVL树的高度平衡，因此需要做平衡化旋转。

被删除的结点P存在以下情况：

1、被删除的结点P有两个孩子

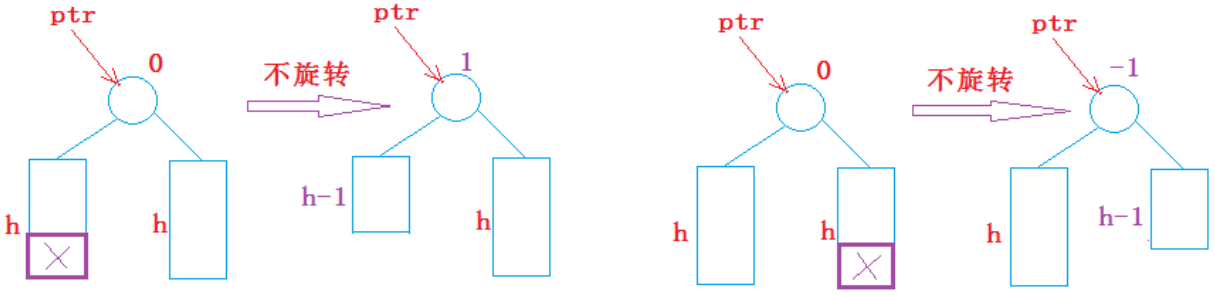
首先搜索P在中序遍历中的直接前驱q(或直接后继)。再把结点q的内容传送给P，问题由删除节点P转移到删除节点q，它是只有一个孩子的结点。

2、被删除的结点P最多只有一个孩子q

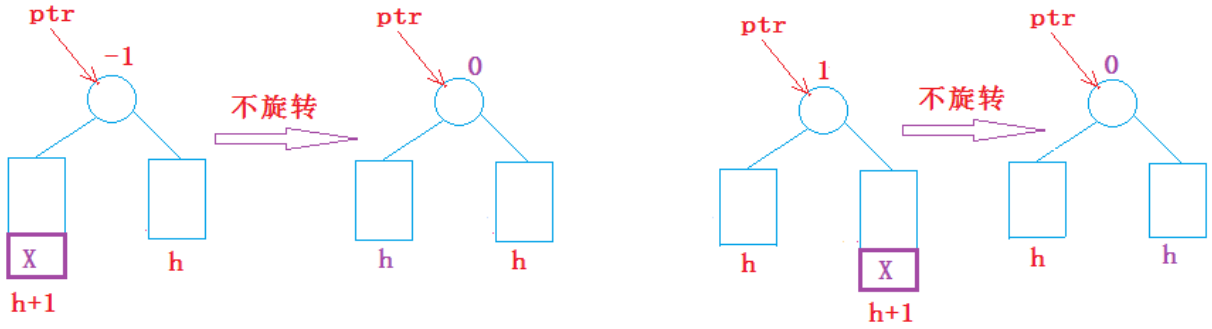
把P的双亲结点Pr中原来指向P的指针该指向q；如果P没有孩子，直接将Pr的相应指针置为NULL。然后将原来以Pr为根的子树的高度减1，并沿Pr通向根的路径反向追踪高度的变化对路径上的各个结点的影响。

考查结点q的双亲结点P，若q是Pr的左孩子，则Pr的平衡因子增加1，否则减少1，根据修改后的Pr的平衡因子值，分三种情况处理：

1、Pr平衡因子原来为0，在其左(或右)子树删除结点后，它的平衡因子增加(减少)1，Pr高度不变，因此从Pr到根所有节点高度均不变，不用调整。

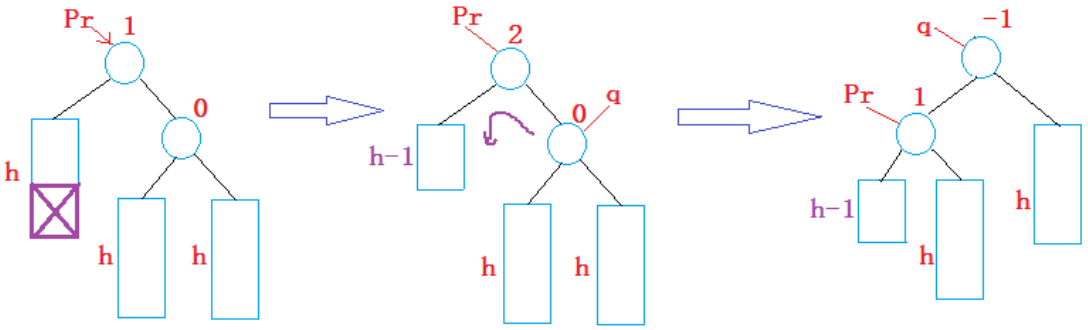


2、Pr原来的平衡因子不为0，且较高的子树被缩短，Pr的平衡因子变成0，此时Pr为根的子树平衡，其高度减1，但需要检查Pr的双亲结点的平衡性。



3、结点Pr的平衡因子不为0，且交矮子树被缩短，则Pr发生不平衡，需要进行平衡化旋转令使其平衡。令Pr较高子树的根为q，根据q的平衡因子，分一下三种情况

a、如果q的平衡因子为0，执行单旋转恢复Pr



b、如果q的平衡因子与Pr平衡因子(正负)号相同，则执行一个单旋转恢复Pr

c、如果q的平衡因子与Pr平衡因子(正负)号相反，则执行一个双旋转恢复Pr