红黑树

五个条件

1.每个节点非黑即红

2.根节点是黑色

3.叶节点 (NIL)是黑色的 //虚拟叶节点时黑色的

4.如果一个节点是红色，则它的两个子节点都是黑色的

5.从根节点出发到所有叶节点路径上，黑色节点数量相同

调整策略

1.插入调整站在`祖父节点`看

2.删除调整站在`父节点`看

3.插入和删除的情况处理一共五种

推论

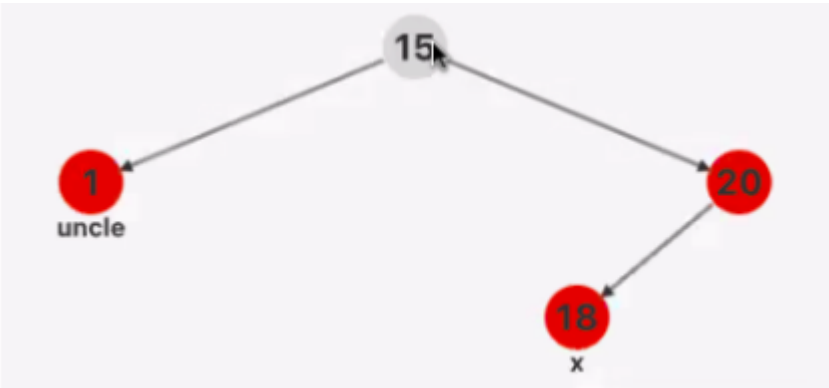
左（右）子树的高度最多是右（左）子树的二倍，即：若H(left) > H(right), 则H(left) <= 2 * H(right) + 1

插入调整的情况

为了去掉两个连续的红色结点

站在祖父结点看父亲和儿子是否冲突（每个插入的结点都是红色的）

情况一



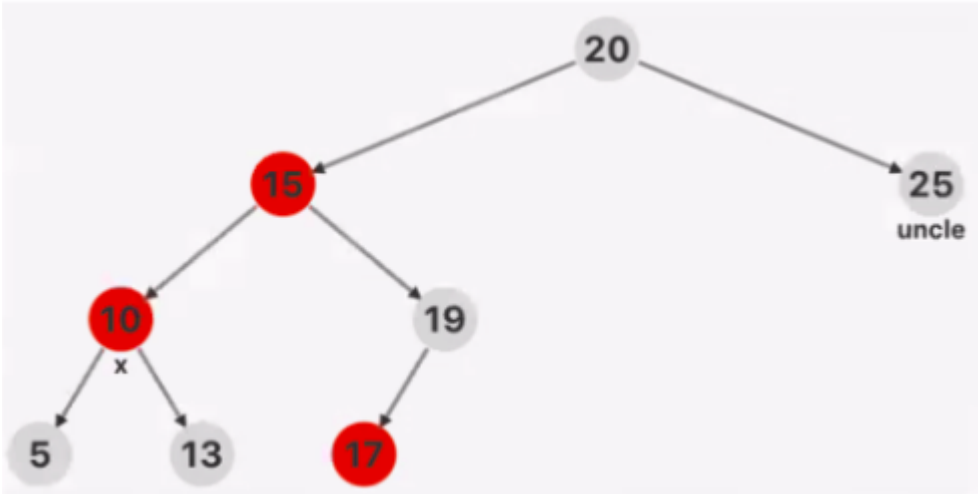
要回溯到祖父结点进行判断是否冲突 要维护红黑树的性质（保证每条路径上的黑色结点数量相同）

发生这种情况的冲突当前结点一定是黑色的

把父节点改为黑色把祖父结点改为红色

处理方法：1和20修改成黑色，15修改成红色（红色上顶）

情况二



黑红红叫红色下沉

红黑黑叫红色上顶

站在祖父结点看右儿子是黑色不是红色

旋转前黑色结点个数保证旋转后黑色结点个数相同

保证可以为路径上提供黑色结点个数相同

处理方法：大右（左）旋，20调整成红色，15调整成黑色（调整策略和AVL树是相同的）

LL，RR，RL，LR型调整和AVL树相同调整后再调色

删除调整的情况

为了去掉双重黑

兄弟结点红色在左边或者在右边 左边大右旋右边大左旋

兄弟结点黑色 孩子都红 都黑 黑红

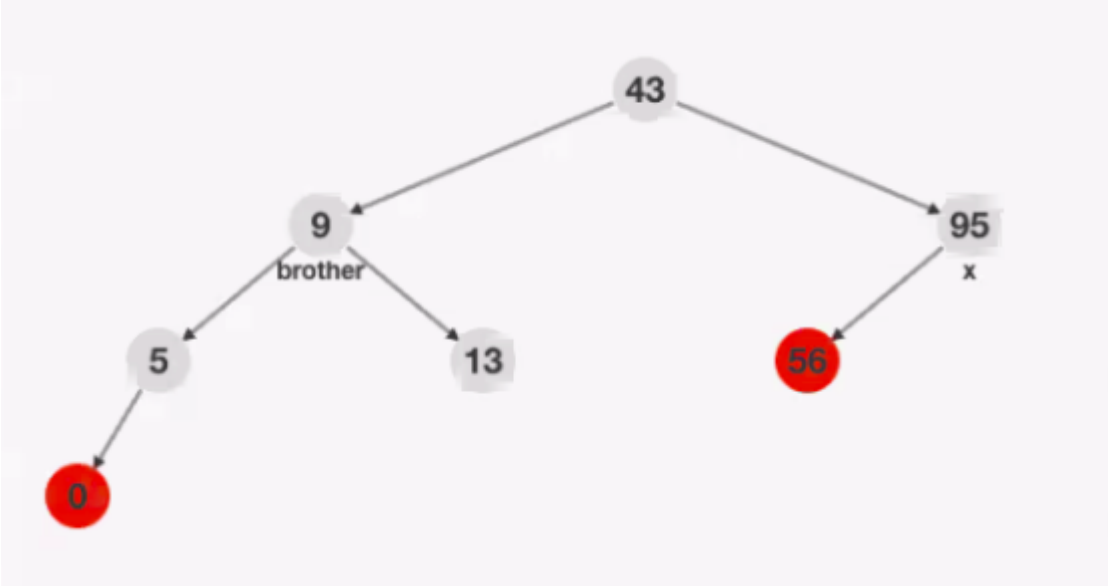
共6种情况

六种情况度为0的黑色或者红色，度为1的黑色或者红色，度为2的黑色或者红色

数值是为了保证排序信息的 颜色是为了保证平衡性的

度	2	2	1	1	0	0
颜色	黑	红	黑	红	黑	红
	直接正常删除	直接正常删除	双重黑NIL把原节点删除	父亲儿子均为黑色不会影响平衡直接删除即可	父节点下一定有两个子孩子将这个结点的虚拟节点变为双重黑色为了去掉双重黑的状态父节点加一重黑兄弟减一重黑	直接删除

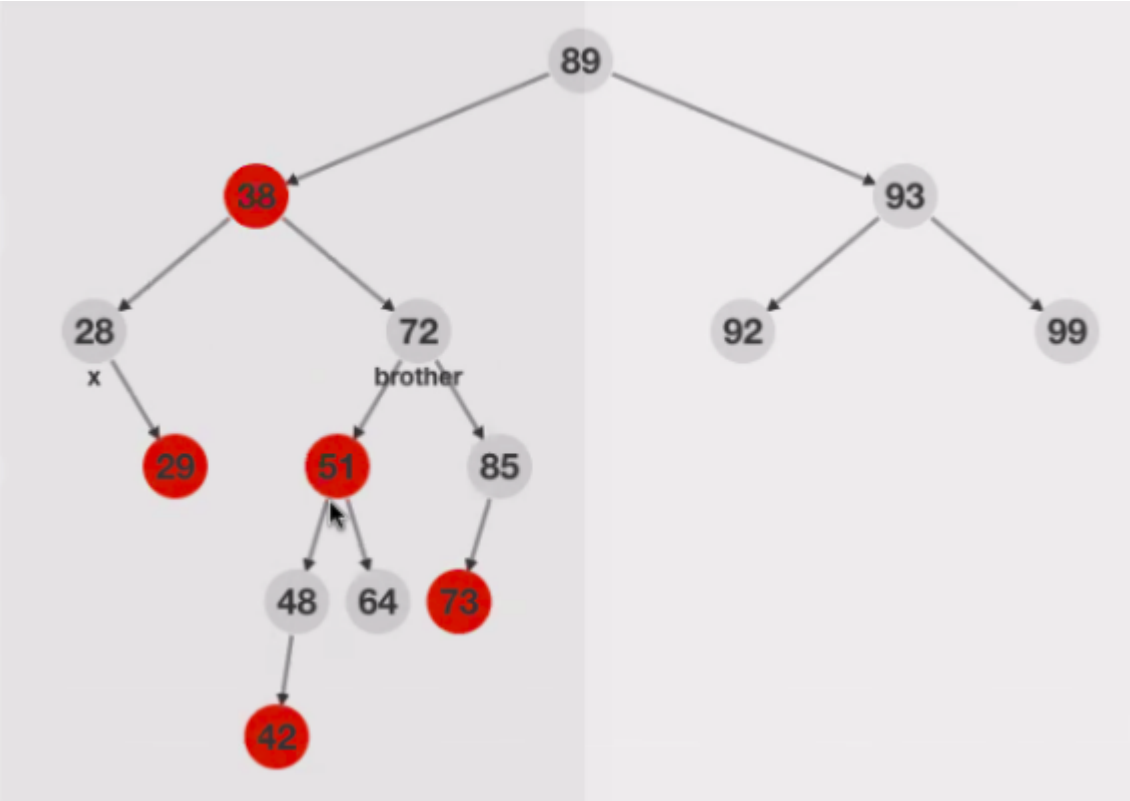
情况一



x为双重黑色brother黑色带2个黑色儿子将兄弟结点改为红色父节点加一重黑色，x减一重黑色

处理方法：brother调整为红色，x减少一重黑色，father增加一重黑色

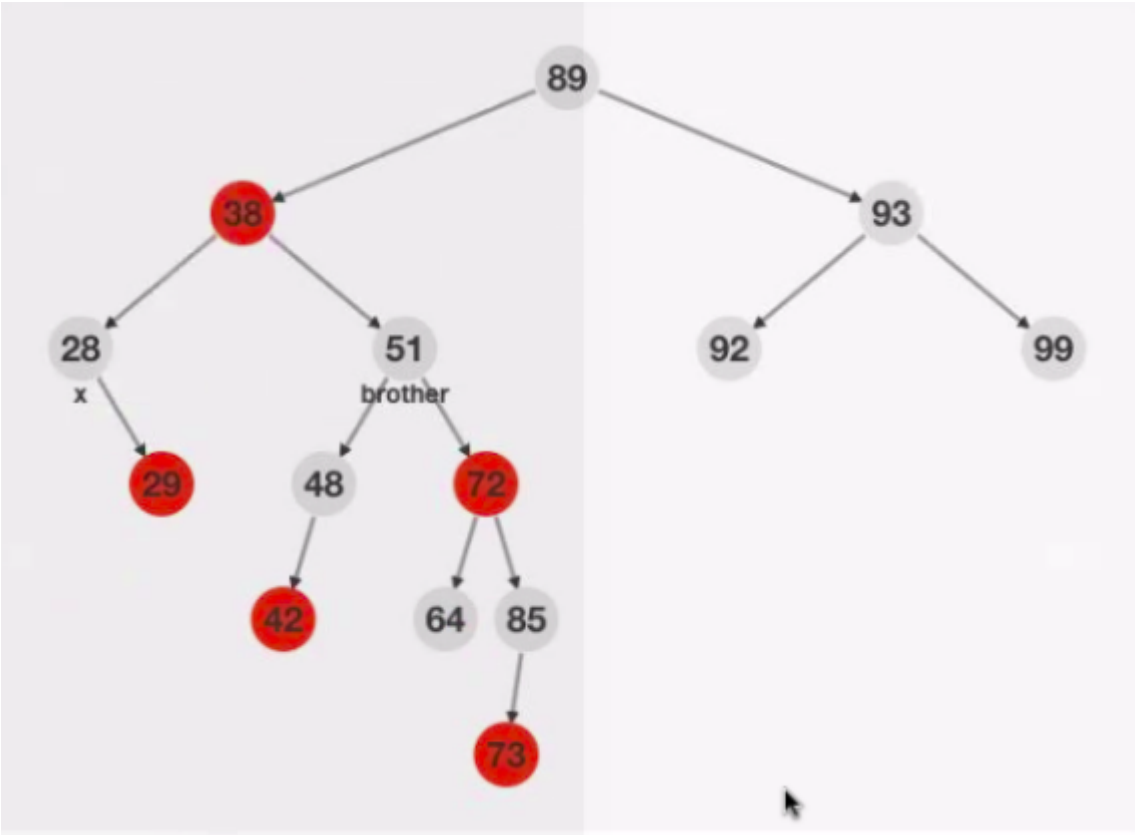
情况二



RL先小右旋改颜色再按RR型改颜色

处理方法：brother右（左）旋，51变黑，72变红，攥成处理情况三

情况三



兄弟结点为黑色单一个孩子结点为红色

RR先大左旋把两课子树都改为黑色根节点改为原来的颜色把双重黑的结点减一重黑

处理方法：father左（右）旋，由于无法确定48的颜色，所以38改为黑色，51改为38的颜色，x减少一重黑色，72改为黑色