

红黑树和2-3树的等价性

那2-3树对应的红黑树是什么样的呢?

红黑树添加

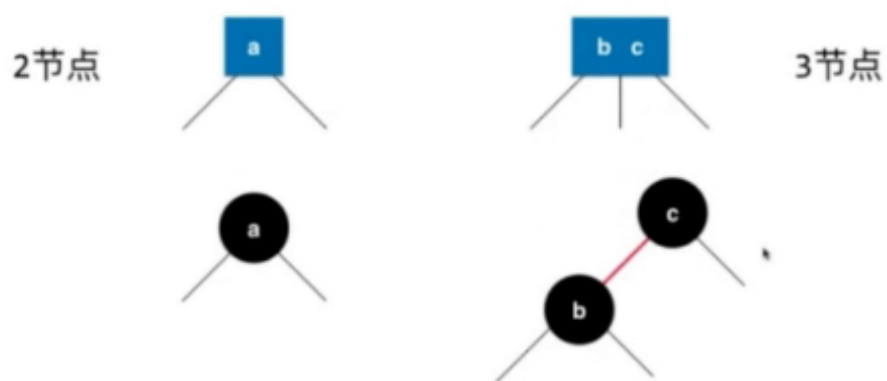
3节点

左旋转

右旋

红黑树和2-3树的等价性

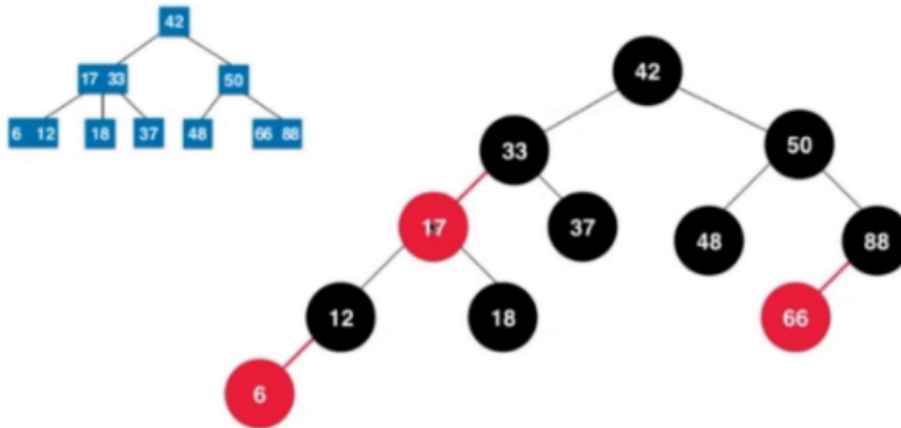
红黑树和2-3树



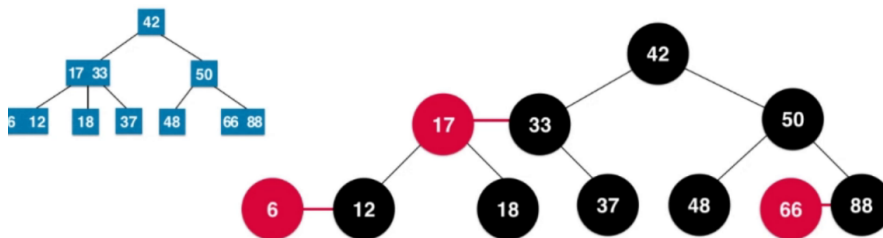
bc 这两个节点在2-3树中是一个并列的关系,但是在红黑树中没有属性表示,所以**b**就标识为红色,所有的红色节点都是左倾斜的

那2-3树对应的红黑树是什么样的呢?

红黑树和2-3树



红黑树和2-3树



红黑树是保持“黑平衡”的二叉树

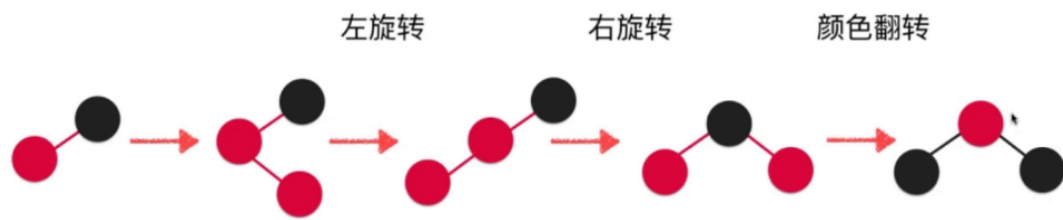
严格意义上，不是平衡二叉树 最大高度： $2\log n$ $O(\log n)$

1. 每一个节点要是红色要么是黑色的
2. 根节点是黑色的。（对应2-3树无论是2节点-黑色 3节点红色-黑色 都是先连接黑色）
3. 如果一个节点是红色那么他的孩子一定是黑色的
4. 从任意一个节点到叶子节点。经过的黑色节点都一样多（每经过一个黑色的节点，就等同于经过了2-3树的某一个节点 2-3树是绝对平衡的）严格意义上 黑平衡不是平衡树

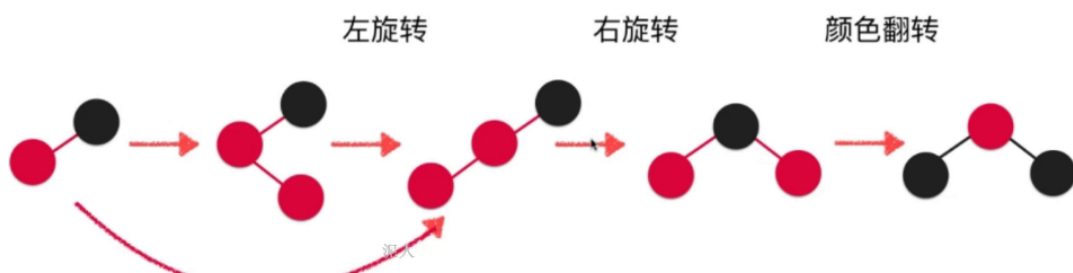
红黑树添加

3节点

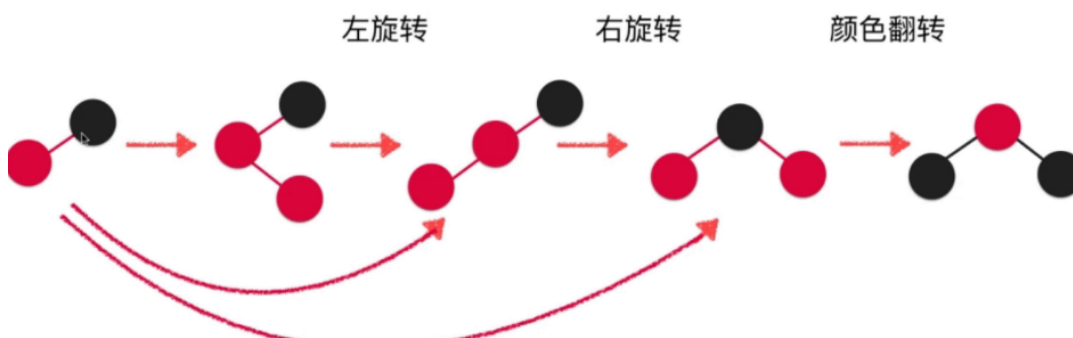
1. 2-3树3节点 新添加的元素在3节点中间 需要左旋转



2. 添加的元素较小

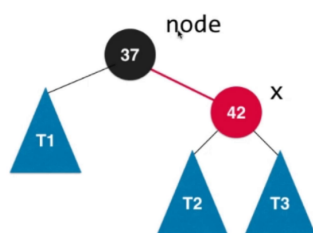


3. 添加的元素最大

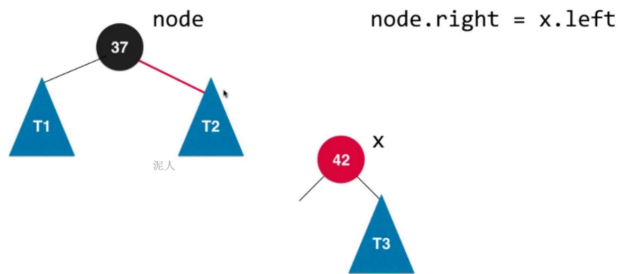


左旋转

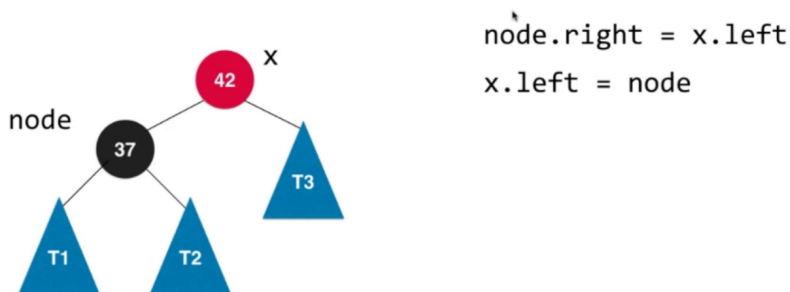
左旋转



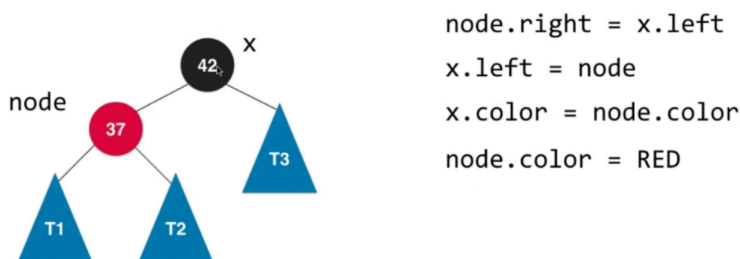
左旋转



左旋转



左旋转



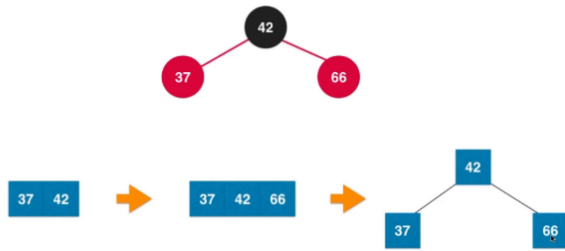
x的颜色是原来的node的颜色.37原来是黑色.所以42的也是黑色.,我们新加入的元素是42他是红色.所以最后 37就是红色

如果是红色,最后也会变成黑色

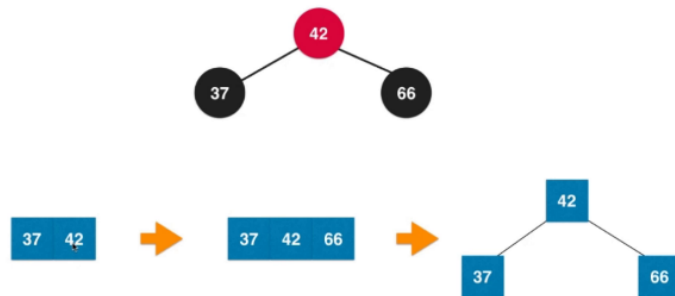
1.如果只有一个节点是黑色的,那么有一个红色的添加到右边会进行左旋转,如果添加到左边的话,不用旋转

2.在2-3树中3节点,相当于红黑树黑化,2-3树种3节点分开之后还要和父节点融合,在红黑树中的表示是变化颜色 如下图

红黑树添加新元素



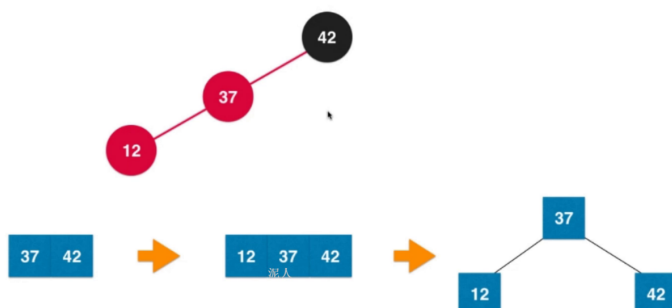
红黑树添加新元素



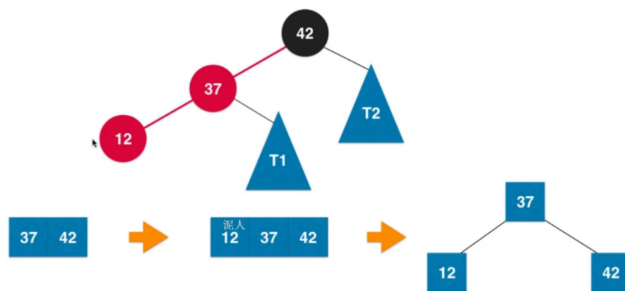
右旋

1. 原来有俩个节点 37 42 现在又来一个12 那这么这个时候需要右旋转
2. $42.left = t1$
3. $37.right = 42$
4. 进行颜色翻转

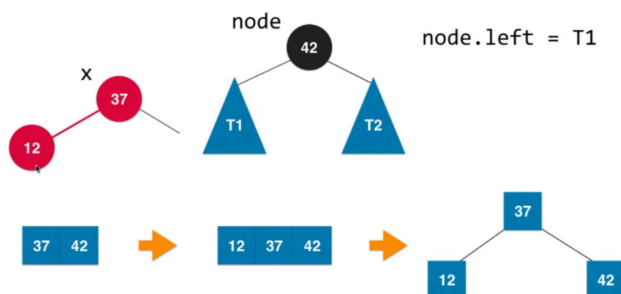
右旋转



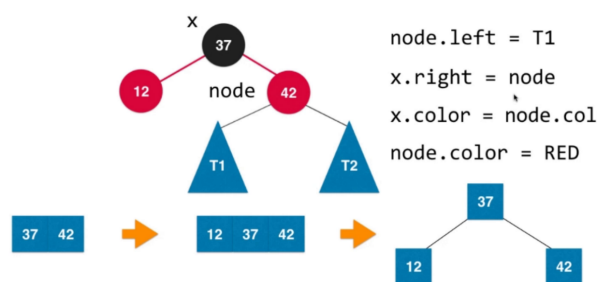
右旋转



右旋转



右旋转

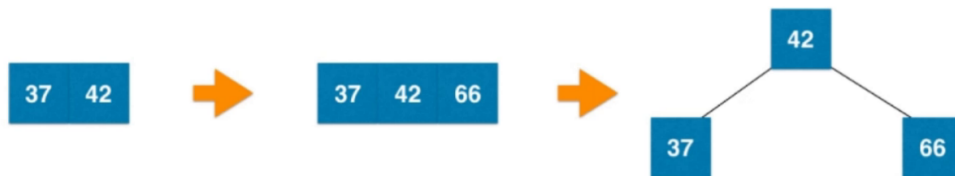
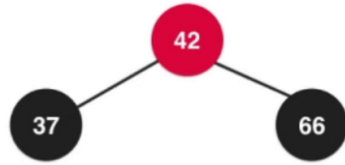


颜色翻转

红黑树添加新元素

颜色翻转

flipColors



- 1.当红黑树添加变成为一个4节点的时候,在2-3树种需要变成一一个的2节点
那么在红黑树中的表示就是都是黑色,,但是变成这3个二节点的父节点还需要和他的父节点融合,所以42是红色的。