

红黑树

胡庆海

主要内容

- ① 红黑树的定义
- ② 红黑树性质
- ③ 红黑树的基本操作
 - 红黑树的查找
 - 红黑树的插入
 - 红黑树的删除
- ④ 参考文献

红黑树的定义

Definition

红黑树： 满足以下五条性质的二叉搜索树称为红黑树

- 1 任意节点的颜色不是红色就是黑色
- 2 根节点颜色必须是黑色
- 3 叶子节点(NIL)颜色必须是黑色
- 4 红色节点的儿子节点必须是黑色
- 5 对于每个节点,从该节点开始到其后代的任何叶子节点路径黑长度相同

红黑树的定义

举例

如下图,树B是红黑树, 而树A不是,因为违反规则4和规则5

¹灰色节点代表节点颜色为黑,红色节点代表颜色为红, 小尺寸节点代表黑色叶子节点, 白色节点代表颜色有多重, 绿色节点代表一个子树

红黑树的定义

举例

如下图,树B是红黑树, 而树A不是,因为违反规则4和规则5

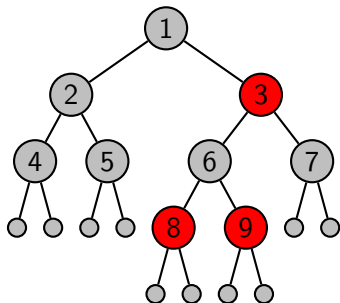


Figure: 树A

¹灰色节点代表节点颜色为黑,红色节点代表颜色为红, 小尺寸节点代表黑色叶子节点, 白色节点代表颜色有多重, 绿色节点代表一个子树

红黑树的定义

举例

如下图,树B是红黑树, 而树A不是,因为违反规则4和规则5

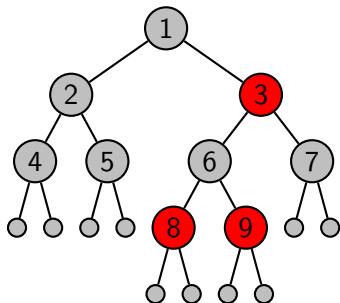


Figure: 树A

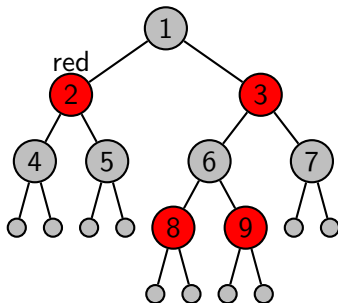


Figure: 树B

1

¹灰色节点代表节点颜色为黑,红色节点代表颜色为红, 小尺寸节点代表黑色叶子节点, 白色节点代表颜色有多重, 绿色节点代表一个子树

红黑树性质

红黑树性质

- 1 任意一个节点其左右两个子树的深度之差不会超过一倍

红黑树性质

- 1 任意一个节点其左右两个子树的深度之差不会超过一倍
- 2 红黑树是平衡树，对于有 n 个节点的二叉树其高度为 $lg n$

红黑树的插入

对于红黑树的插入，我们首先使用BST的插入算法找到插入点并插入，然后设置其颜色为红色，再根据其具体情况进行修改，主要情况有以下四种

- 1 插入节点的父节点是黑色
- 2 插入节点的父节点是红色，叔叔节点也是红色
- 3 插入节点的父节点是红色，叔叔节点也是黑色，插入节点是父节点的右子
- 4 插入节点的父节点是红色，叔叔节点也是红色，插入节点是父节点的左子

插入情况1

由于此时没有破坏红黑树的任何性质，所以算法结束

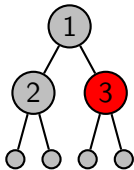


Figure: 插入节点3

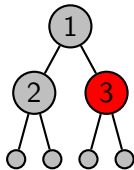


Figure: 调整之后

父节点和叔叔节点都是红节点，那么爷爷节点肯定是黑节点，这样只需要把父节点和叔叔节点染成黑色，爷爷节点染成红色，问题就变成调整爷爷节点了

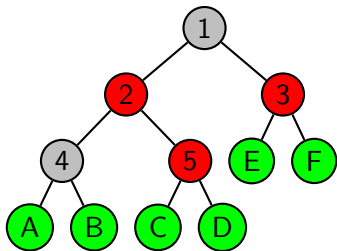


Figure: 插入节点5

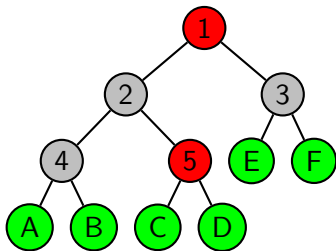


Figure: 调整之后,继续调整1

插入情况3

父节点是红节点，叔叔节点是黑节点，且插入节点是父节点的右子，那么可以以父节点为旋转点做左旋，转化为情况4

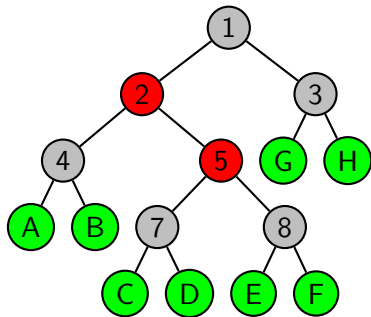


Figure: 插入节点5

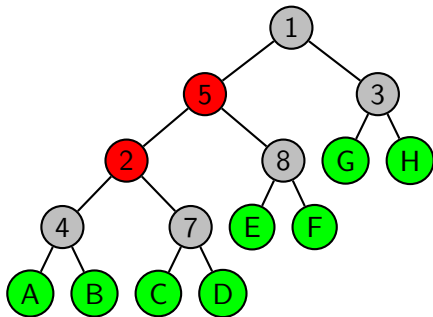


Figure: 调整之后,继续调整2

插入情况4

父节点是红节点，叔叔节点是黑节点，且插入节点是父节点的左子，那么可以以爷爷节点为旋转点右旋并交换父节点和爷爷节点的颜色，最终调整完毕，退出算法

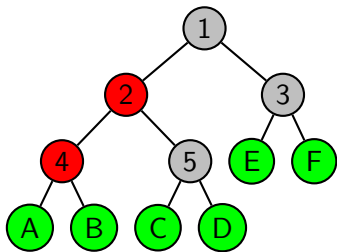


Figure: 插入节点4

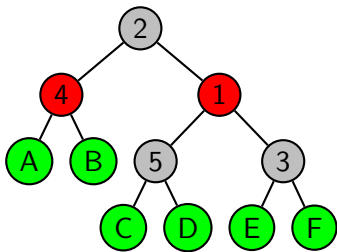


Figure: 调整之后

直接将此节点颜色设为黑色

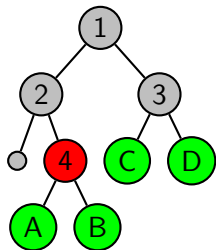


Figure: 删除节点2

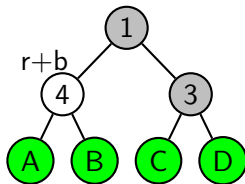


Figure: 调整节点4

删除情况2

同样，直接将此节点颜色设为黑色

删除情况3

当前节点是B+B,兄弟节点是红节点,则以父节点为旋转点左旋,并且交换父节点和兄弟节点的颜色,将问题转化为后面的情况

删除情况3

当前节点是B+B,兄弟节点是红节点,则以父节点为旋转点左旋,并且交换父节点和兄弟节点的颜色,将问题转化为后面的情况

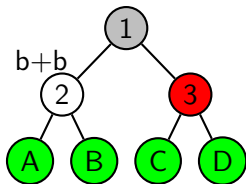


Figure: 调整节点2

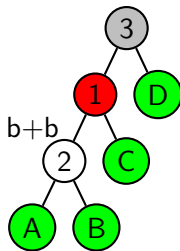


Figure: 调整之后, 继续调整节点2

删除情况4

当前节点是B+B,兄弟节点是黑节点,且兄弟节点的孩子都是黑节点,这种情况下可以把兄弟节点换成红色,那么左右两边都会缺一个黑色,则可把节点多的那重黑色转移给父节点,问题就变成父节点的问题了

删除情况4

当前节点是B+B,兄弟节点是黑节点,且兄弟节点的孩子都是黑节点,这种情况下可以把兄弟节点换成红色,那么左右两边都会缺一个黑色,则可把节点多的那重黑色转移给父节点,问题就变成父节点的问题了

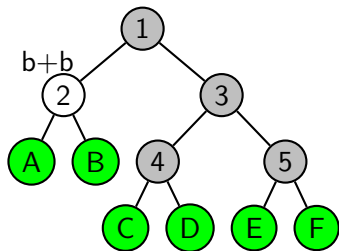


Figure: 调整节点2

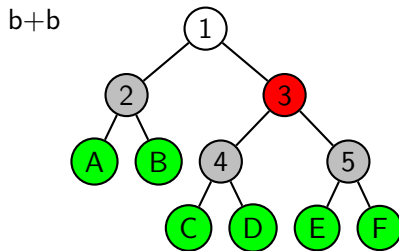


Figure: 调整之后，继续调整节点1

删除情况5

当前节点是B+B,兄弟节点是黑节点,且兄弟节点的孩子左子是红,右子是黑,这种情况下可以右旋它的兄弟节点,把问题转化为情况6

删除情况6

当前节点是B+B,兄弟节点是黑节点,且兄弟节点的孩子右子是红,这种情况下可以左旋父节点,然后交换父节点和兄弟节点的颜色,并把兄弟节点的右子涂成黑色,则问题得以解决,退出算法

Figure: 调整之后，算法结束

参考文献

- [1] http://blog.csdn.net/v_july_v/article/details/6105630
- [2] 算法导论
- [3] STL源码剖析

致谢

Thanks