红黑树（Red-Black tree）：局部平衡的二叉查找树。C++ STL中的set、map以及Linux虚拟内存的管理，都是通过红黑树去实现的。

红黑树的特性：

1. 每个结点或者是黑色，或者是红色；
2. 根结点是黑色的。
3. 如果一个结点是红色的，则它的子结点必须是黑色的；如果一个结点是黑色的，则其子结点可以是黑色，也可以是红色。
4. 红黑树中含有扩充的外部结点（NULL结点）是黑色的
5. 任何结点到其子孙的外部结点的路径上必须包含相同数目的黑色结点。

参考<http://www.cnblogs.com/skywang12345/p/3245399.html#aa2>

<http://blog.csdn.net/v_JULY_v/article/details/6105630>



NIL为扩充的外部结点。

红黑树中祖父，父亲，叔叔结点的概念是为了在插入修复和删除修复时更好的进行着色修改，和一般二叉查找树中的左孩子和右孩子不同。

例：程序red\_black\_tree\_test

将nullptr替换成null\_node

https://my.oschina.net/bgbfbsdchenzheng/blog/495639

找个复杂一点的红黑树，查看其结点删除时，平衡条件的变化。要在删除时将Case1，Case2，Case3和Case4的情况都测试一遍。插入时也要测试一遍。