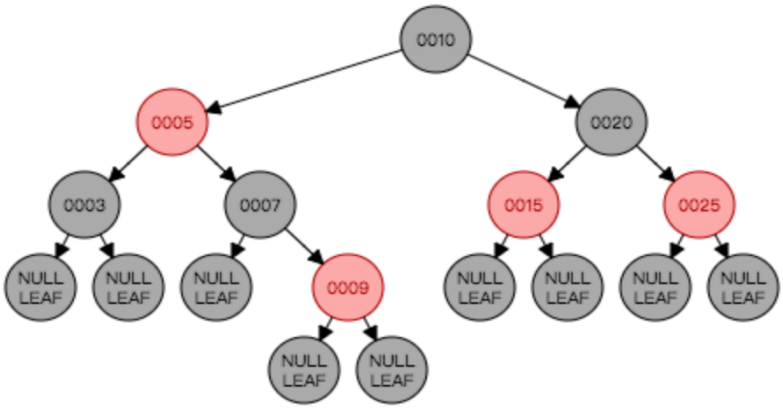
Xinyu Jiang

03/16/2018

Assignment 7

Question 1: Does inserting a node into a red-black tree, re-balancing, and then deleting it result in the original tree?

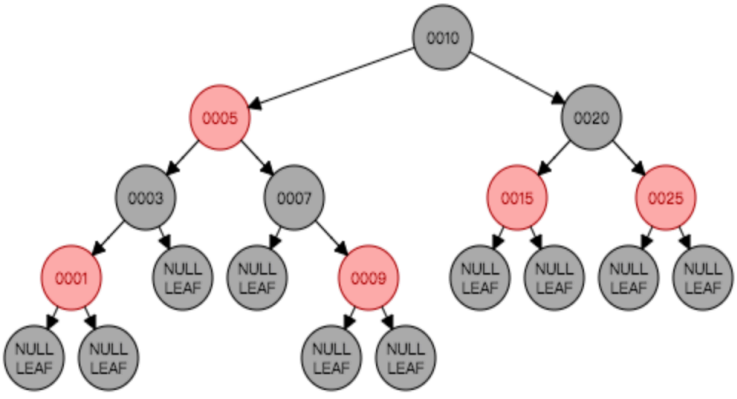
Original Tree



Ex1: Inserting a node

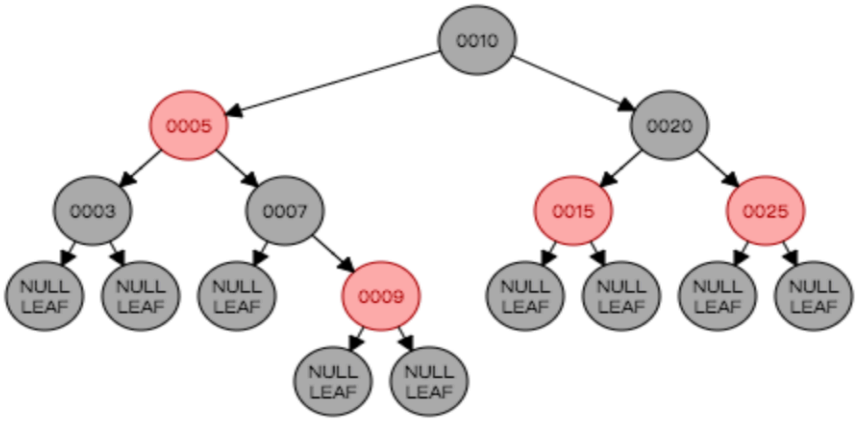
Insert 1: It goes left of the node 3 and the color is red

After Insert:



Re-balance: Because it has no children, after insertion, no need to rebalance the tree, so the tree has already balanced in the picture above.

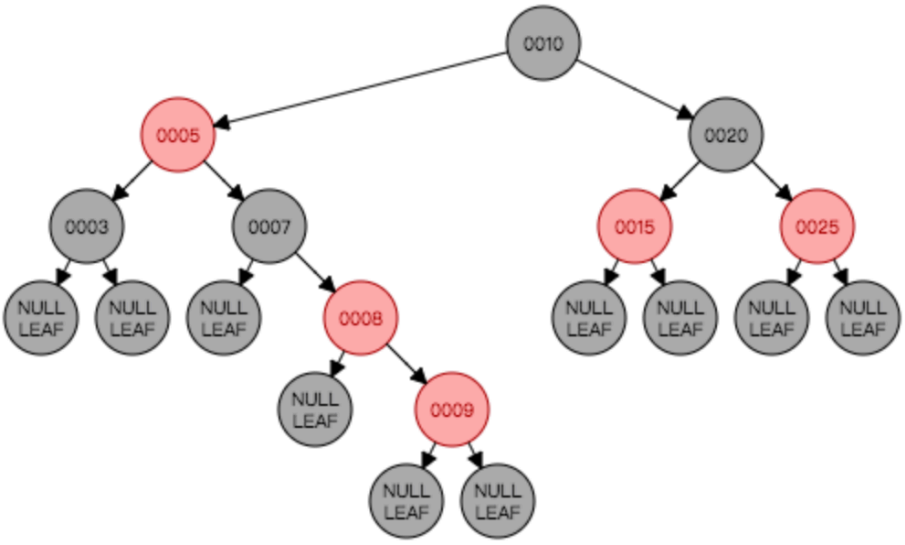
Delete 1: Because 1 is smaller than 10, go to the left, and it is smaller than 5, go to left again, it is small than 3, go to the left, then find 1 and delete it, after delete.



Since the tree is identical with the original one

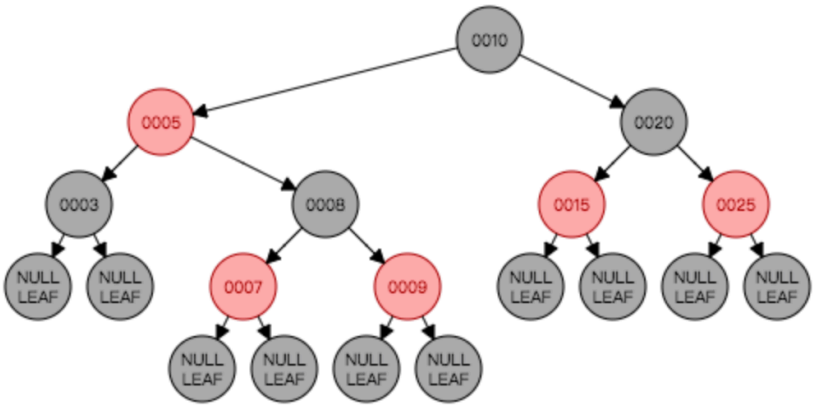
Conclusion, for case 1, Inserting a node into a red-black tree, re-balancing, and then deleting it result in the original tree.

Ex2: Inserting 8



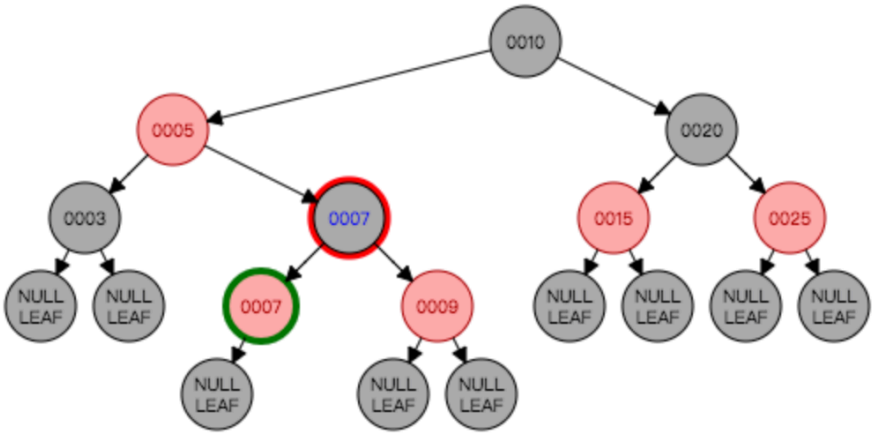
Since “node 9” is red and “node 8”is also red It violates the rule of red black tree. Therefore, in order to balance it, we need to do the right rotate first.

After right rotation: change the color of “Node 8”from red to black, and “Node 7” from black to red. To re-balance it

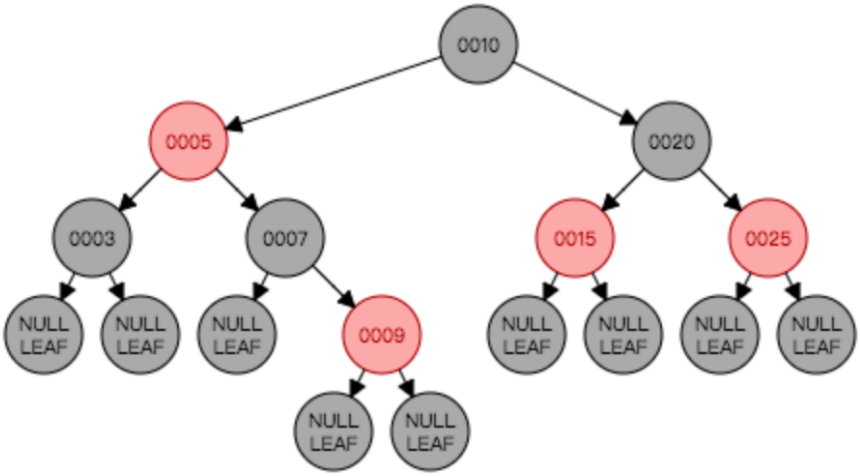


Delete 8:

After find 8, find largest node from left subtree, change “Node 8”value from 8 to 7



Then delete “Node 7” with color red. Then connect the black “Node 7” to “NullNode”.



Since the picture is identical with original.

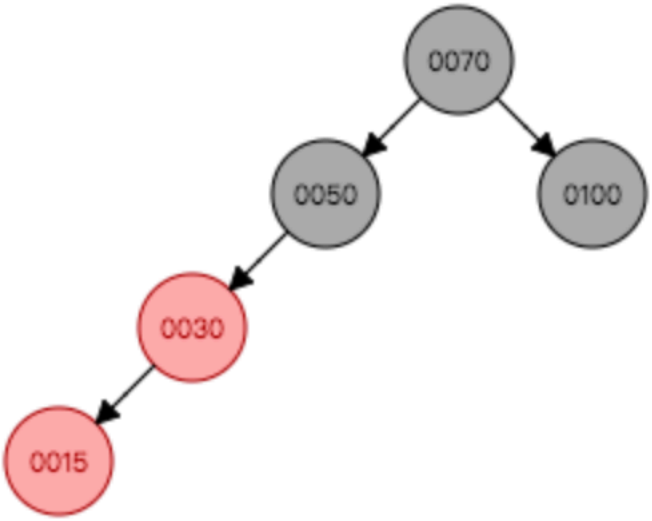
Conclusion, for case 2, Inserting a node into a red-black tree, re-balancing, and then deleting it result in the original tree.

Ex3:

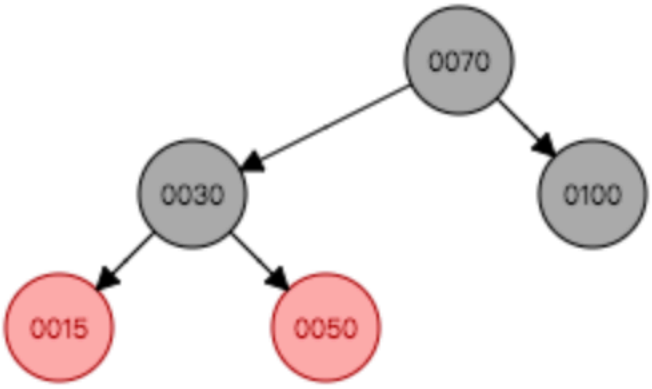
Original tree:

Inserting 15

After inserting without re-balancing:

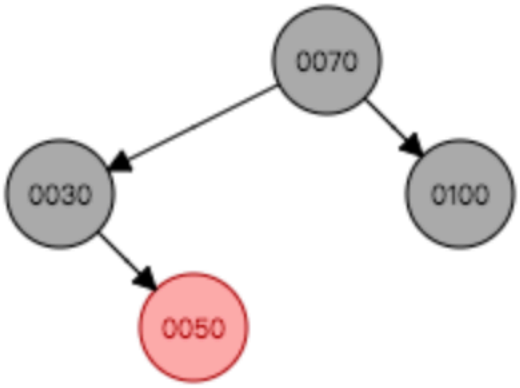


Since “Node 30” and “Node 15” color is both red, it violates the rule of red black tree, we need to do a right rotation then change “Node 30” color from red to black and “Node 50” from black to red to balance it.



Deleting 15:

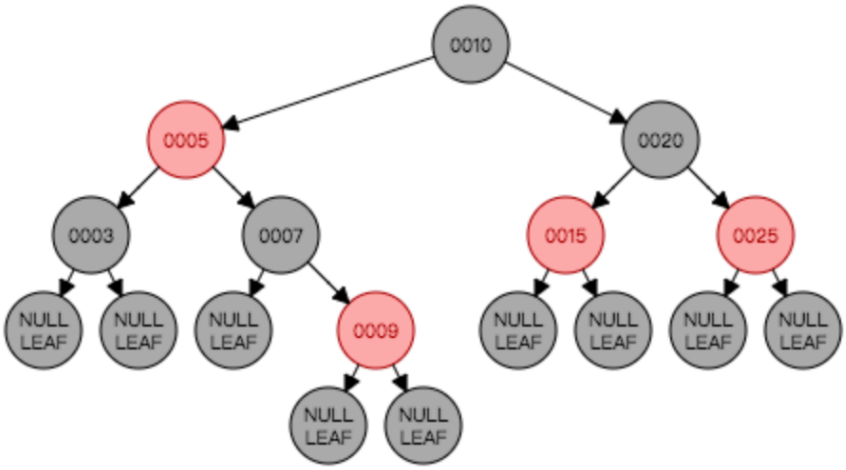
Since 15 is and it has no child, we could just delete it.



Compare to the original tree, the tree is not identical with the original.
Conclusion, for case 3, Inserting a node into a red-black tree, re-balancing, and then deleting it does not result in the original tree.

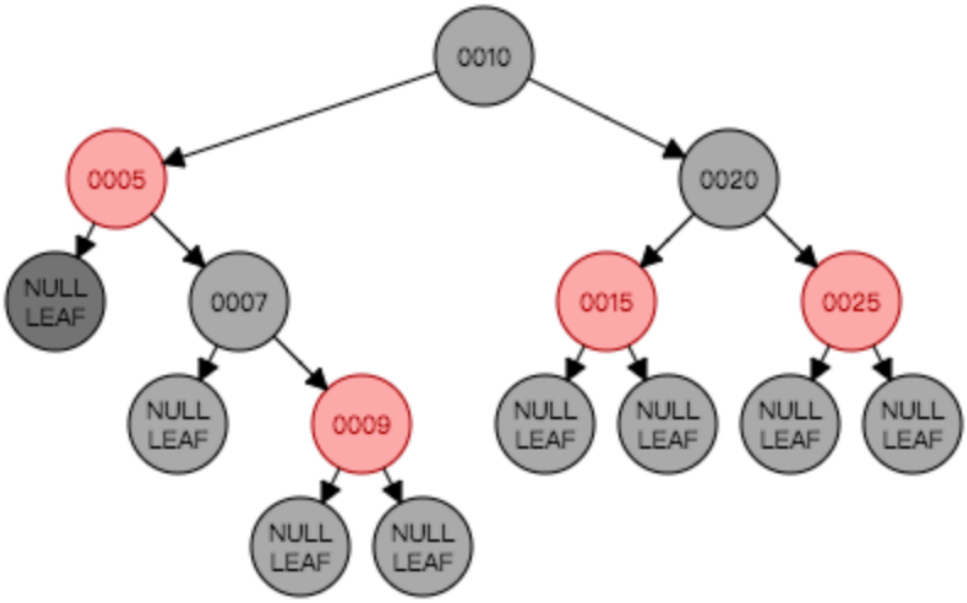
Conclusion for question 1: Inserting a node into a red-black tree, re-balancing, and then deleting it does not always result in the original tree (could change the original tree).

Question 2: Does deleting a node with no children from a red-black tree, re-balancing, and then re- inserting it with the same key always result in the original tree?
Original Picture



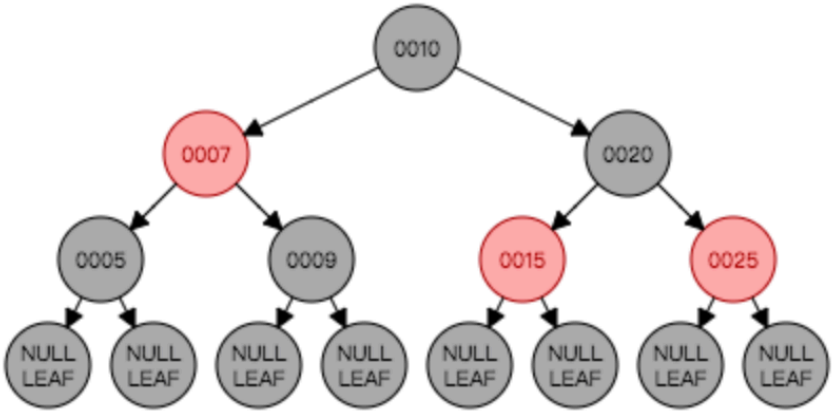
Ex1: Delete “Node 3” which has no children

After deleting without re-balancing



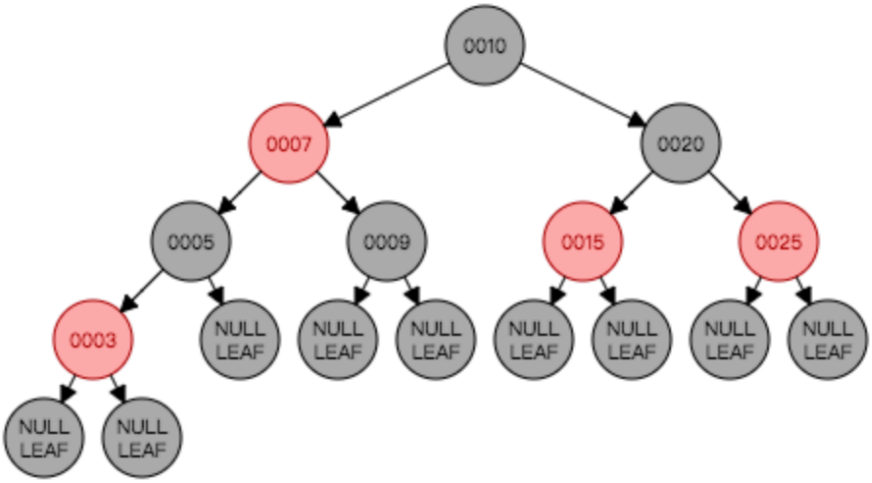
Since the node color is black, the tree need to a right rotation to balance it. After right rotation, change “Node 7” color from black to red and “Node 9” from black to red.

After rotation:



Re-insert “Node 3”

After insertion:



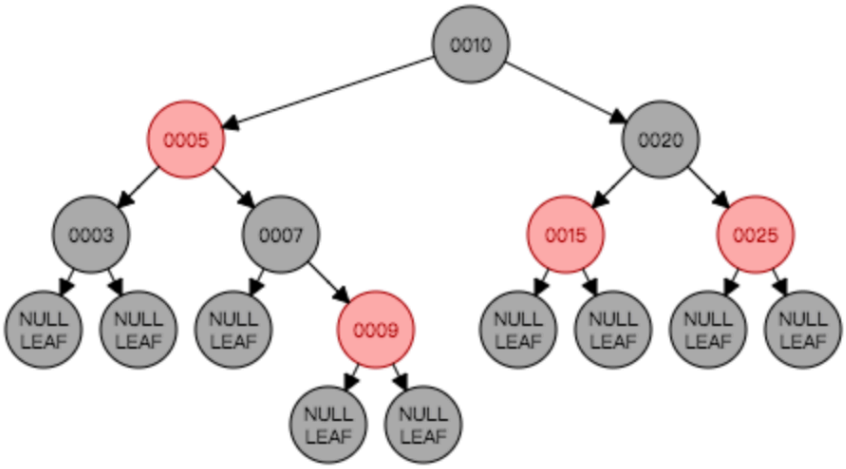
Compare the tree with original it is not identical.

Conclusion for case 1: deleting a node with no children from a red-black tree, re-balancing, and then re- inserting it with the same key does not always result in the original tree.

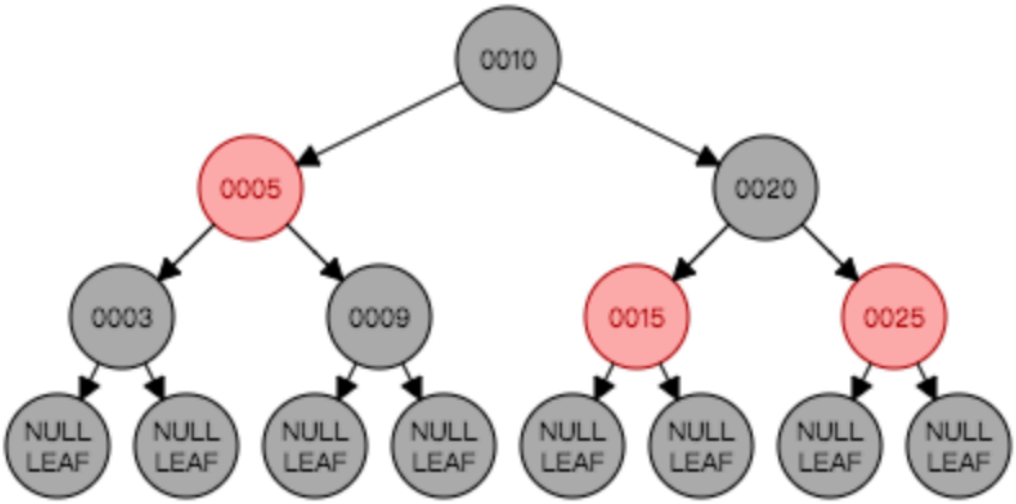
Case 2:

Deleting 7 which has one child

Original Tree:

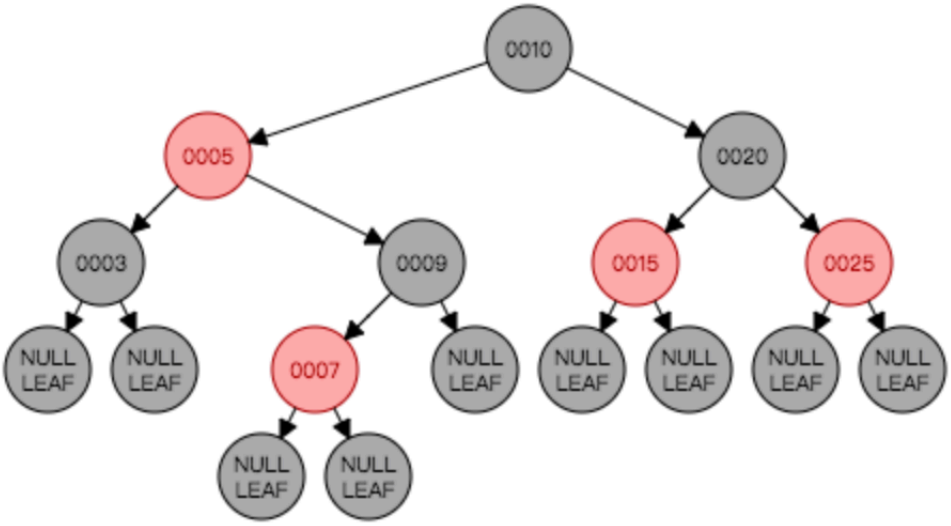


After delete:



Since “Node 7”do not have a left child, we could just delete it and connect “Node 5” to “Node 9”.

Re-inserting 7:

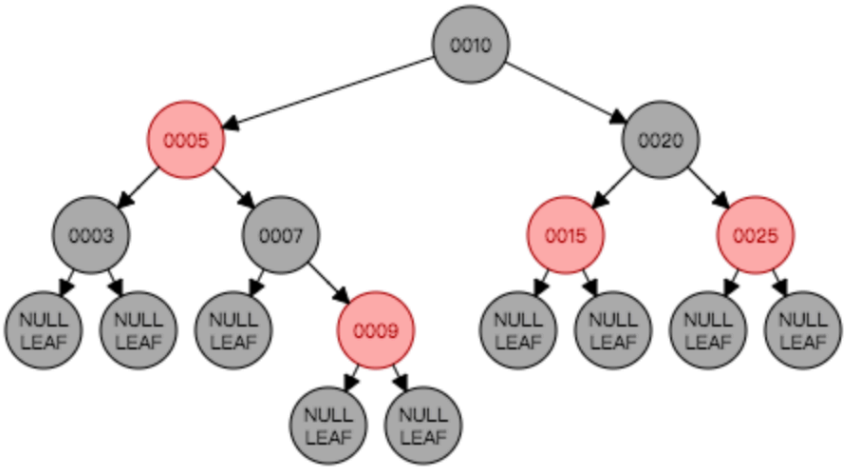


Compare the tree with original it is not identical.

Conclusion for case 2: deleting a node with one child from a red-black tree, re-balancing, and then re- inserting it with the same key does not always result in the original tree.

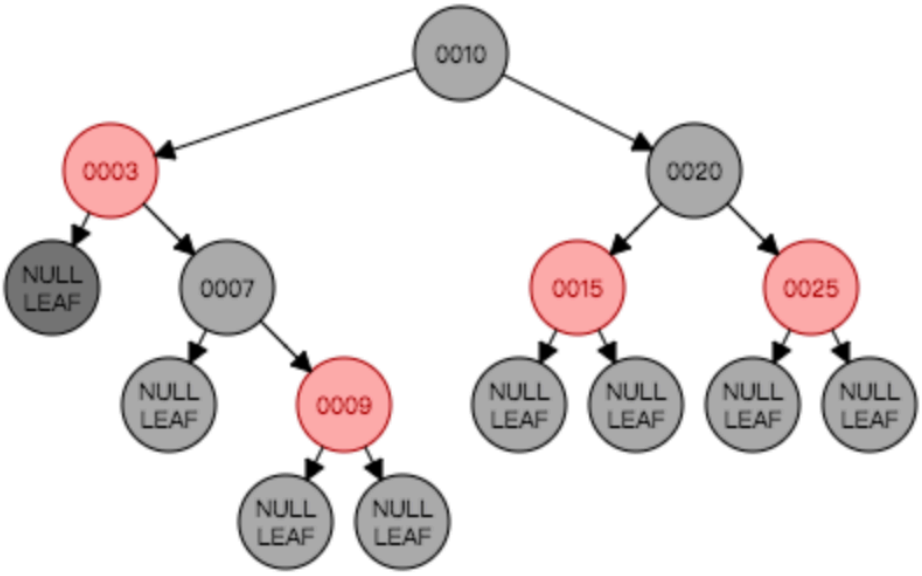
Ex 3: Delete 5 which has two children

Original Tree:

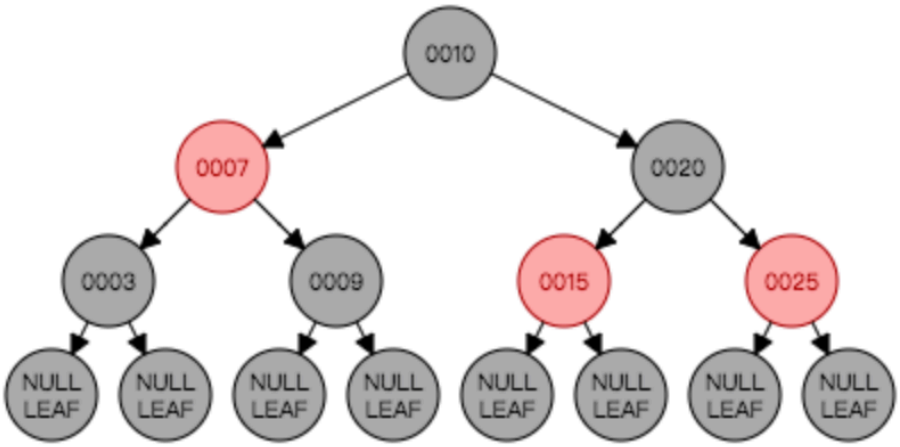


Deleting:

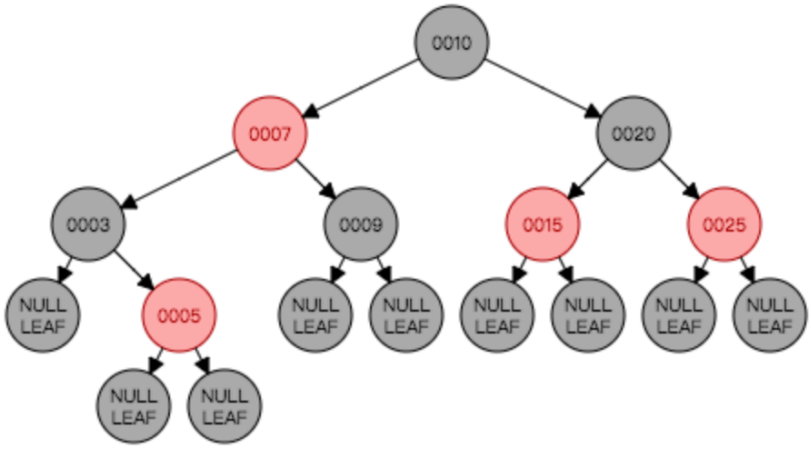
Copy largest left subtree to “Node 5”, then delete it.



Since it violates the rule that all paths need to have equal number of black node, the tree need to do a left rotation to balance it. Then change the color “Node 7” from black to red and the color “Node 9” from red to black.



Re-inserting 5:



Compare the tree with original it is not identical.

Conclusion for case 3: deleting a node with two children from a red-black tree, re-balancing, and then re- inserting it with the same key does not always result in the original tree.

Conclusion for question 2: deleting a node from a red-black tree, re-balancing, and then re- inserting it with the same key does not result in the original tree.