

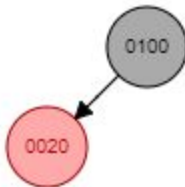
Insert:	100	20	60	50	30
Insert:	40	90	45	25	65
Delete:	90	20	60	45	50
Delete:	100	40	25	65	30

1. Insert: 100



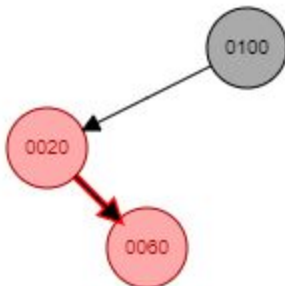
*just insert, because it is the first node, set it to black

2. Insert: 20

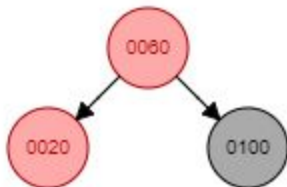


*since $20 < 100$, put 20 as left of node 100, set 20 as red, since its parent is black, no problem

3. Insert: 60



*since $60 < 100$ and $60 > 20$, put 60 as right of node 20, set 60 as red

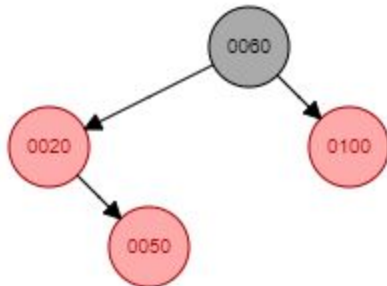


*Node and parent is red, Node is right child, parent is left child, therefore double rotate right the tree

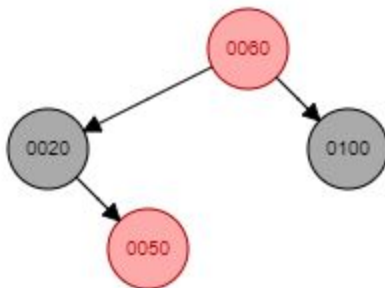


*recolor

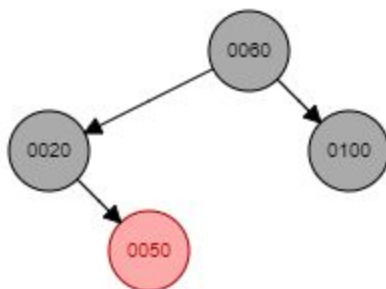
4. Insert: 50



*since $50 < 60$ and $50 > 20$, put 50 as right of node 20, set 50 as red

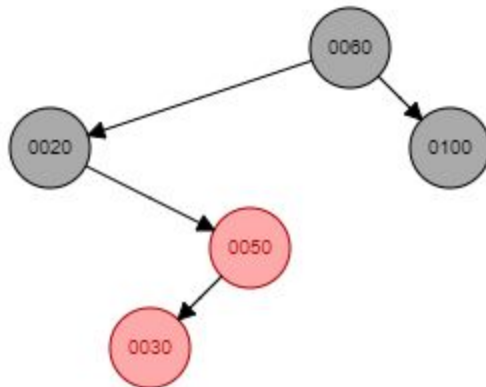


*uncle of 50 is also red, hence push the black color to 20 and 100

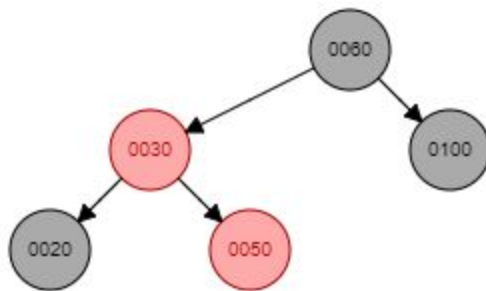


*set grandparent as black

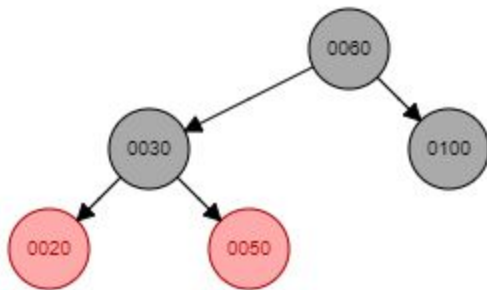
5. Insert: 30



*since $30 < 60$, $30 > 20$, $30 < 50$, put 30 as left of node 50, set 30 as red

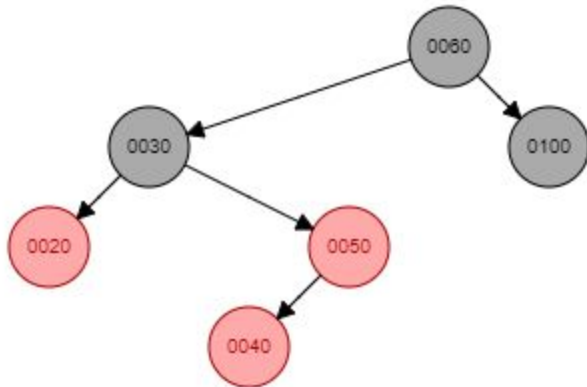


*Node and parent is red, Node is left child, parent is right child, therefore double rotate left the tree.

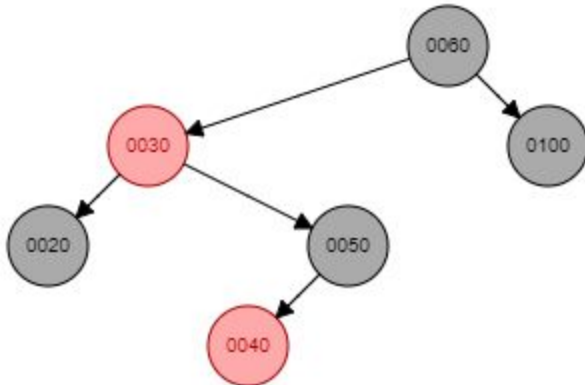


*push the red color to 20 and 50, set 30 as black

6. Insert: 40

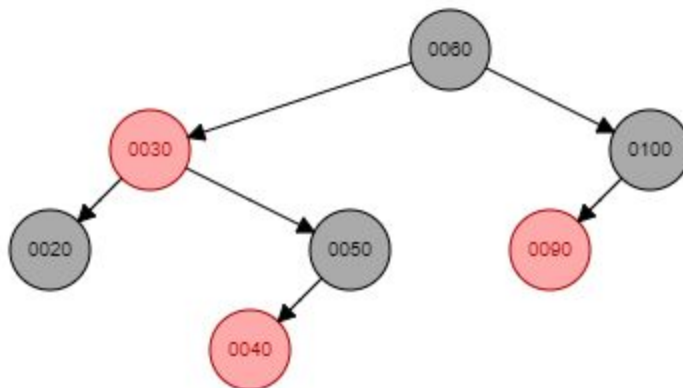


*since $40 < 60$, $40 > 30$, $40 < 50$, put 40 as left of node 50, set 40 as red



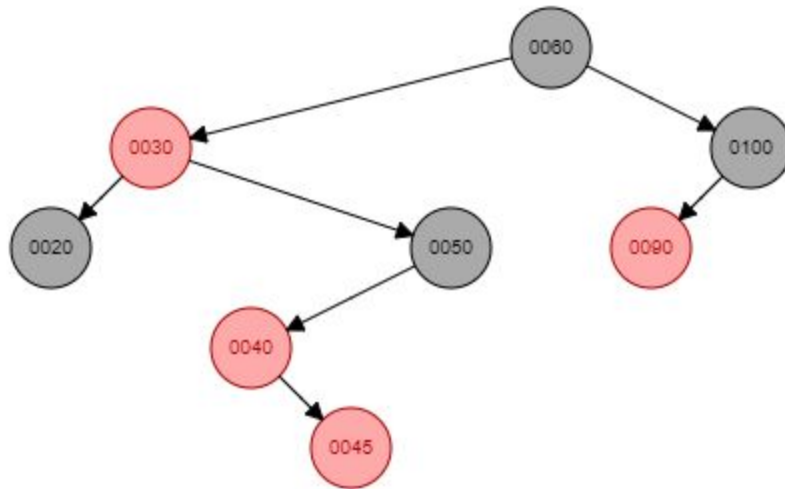
*uncle of 40 is also red, hence push the black color to 20 and 50

7. Insert: 90

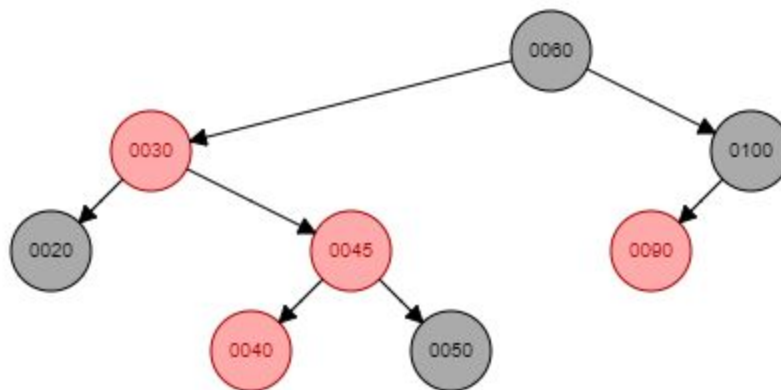


*since $90 > 60$, $90 < 100$, put 90 as left of node 100, set 90 as red

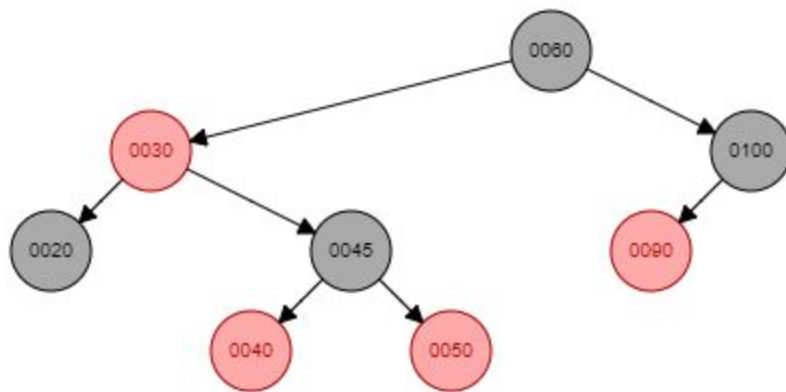
8. Insert: 45



*since $45 < 60$, $45 > 30$, $45 < 50$, $45 > 40$, put 45 as right of node 40, set 45 as red

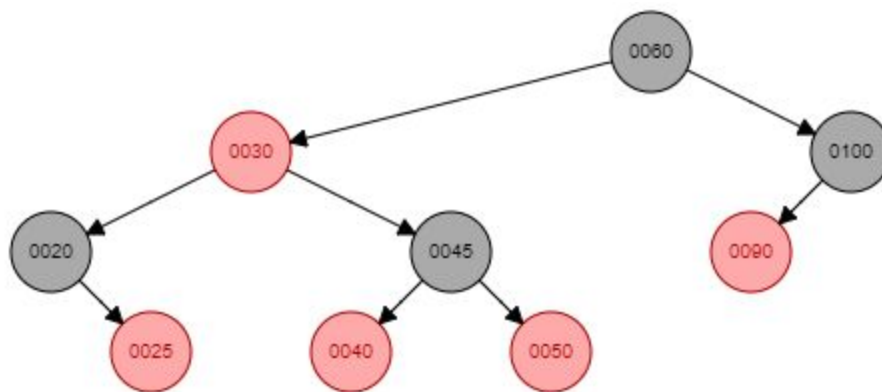


*Node and parent are red, Node is right child, parent is left child, therefore double rotate right the tree.



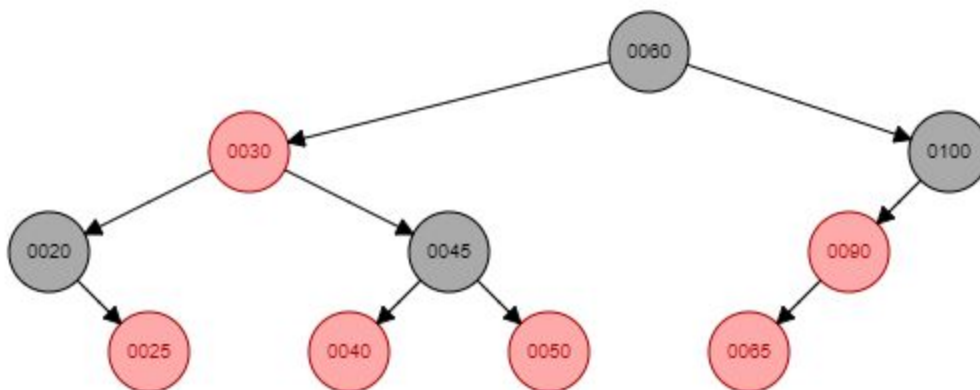
*push red to 40 and 50, set 45 as black

9. Insert: 25

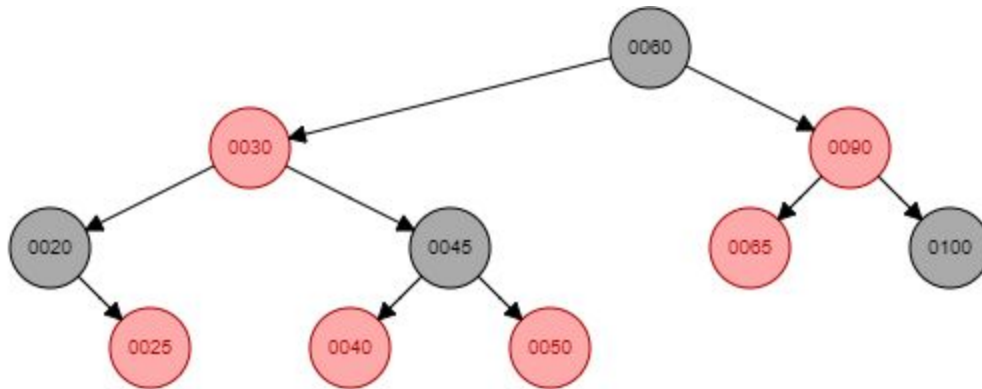


*since $25 < 60$, $25 < 30$, $25 > 20$, put 25 as right of node 20, set 25 as red

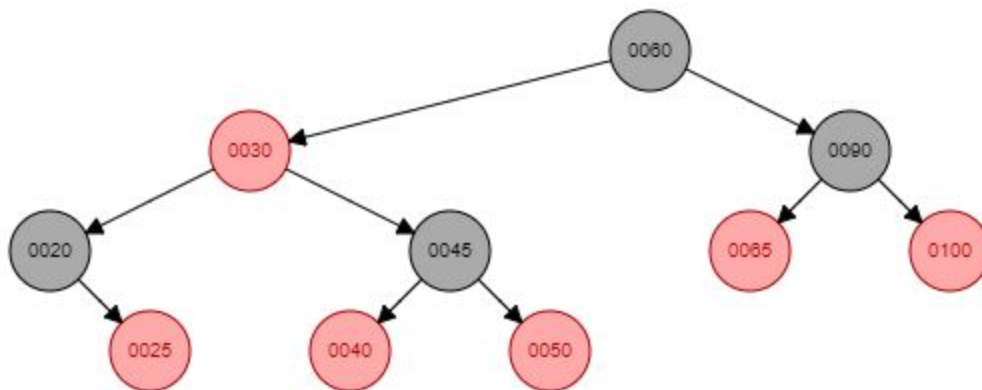
10. Insert: 65



*since $65 > 60$, $65 < 100$, $65 < 90$, put 65 as left of node 90, set 65 as red



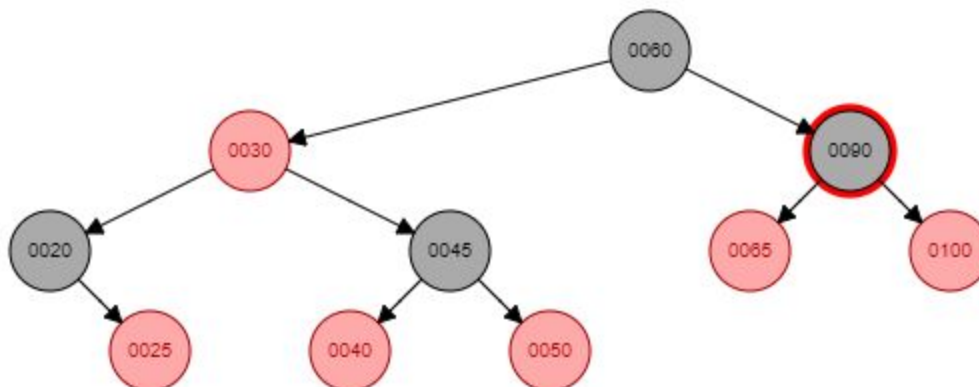
*Node and parent are red, Node is left child, parent is left child, therefore single rotate right the tree



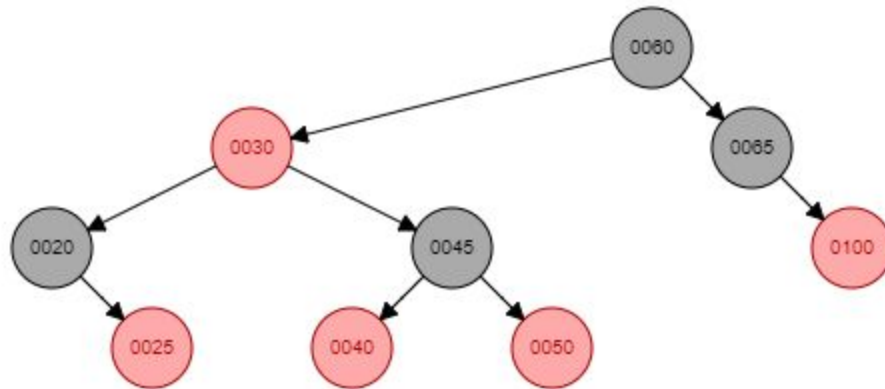
*push red to 65 and 100, change 90 to black

11. Delete: 90

*traverse to find 90



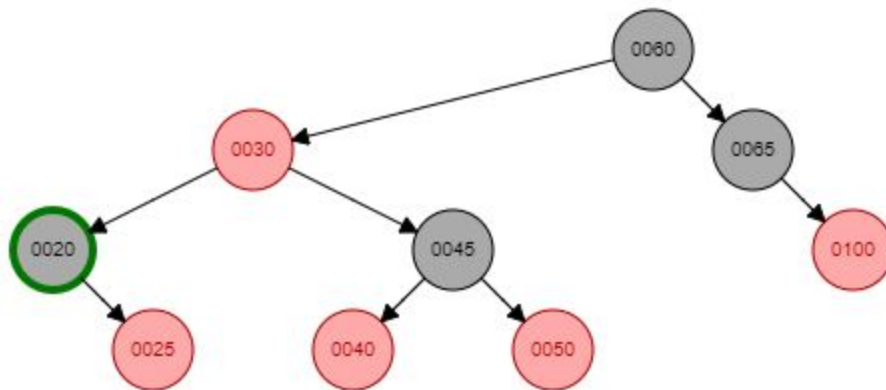
*found 90, but 90 has children



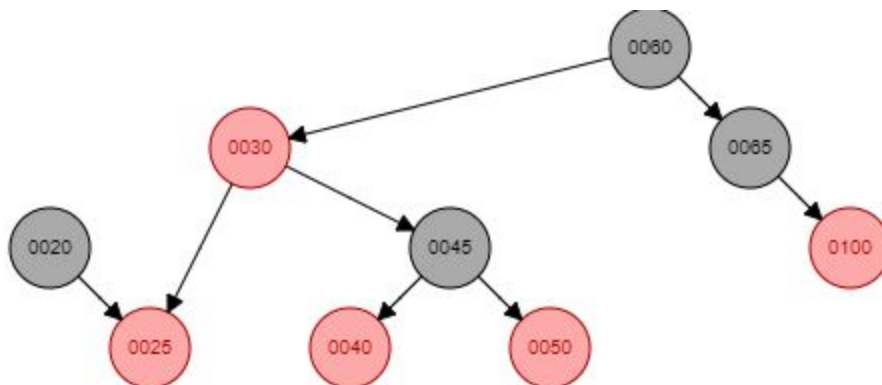
*traverse the left side, because 65 has no child, 65 will replace 90, then we delete node 65

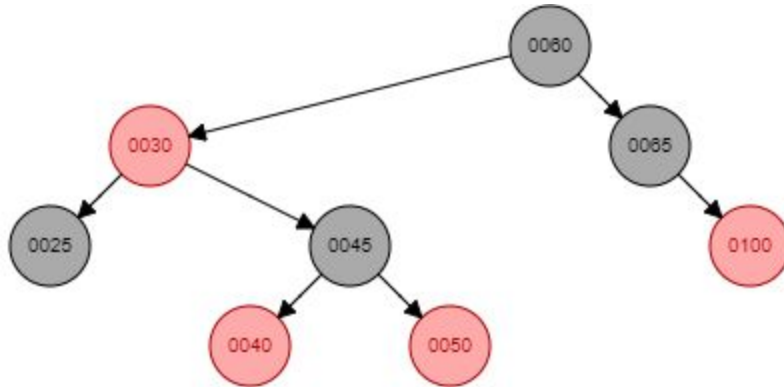
12. Delete: 20

*traverse to find 20



*found 20, but 20 has children

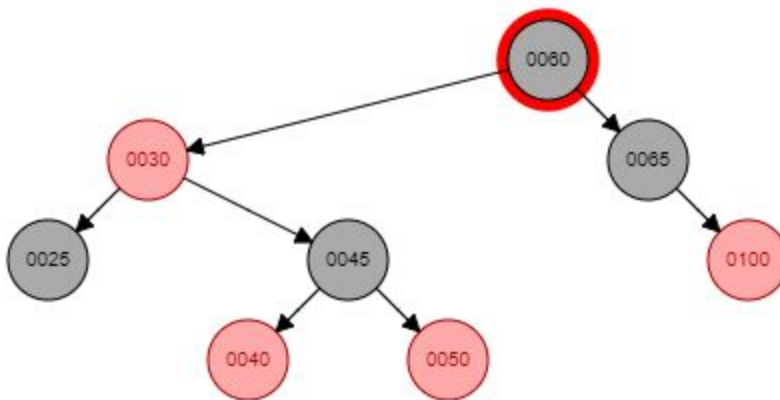




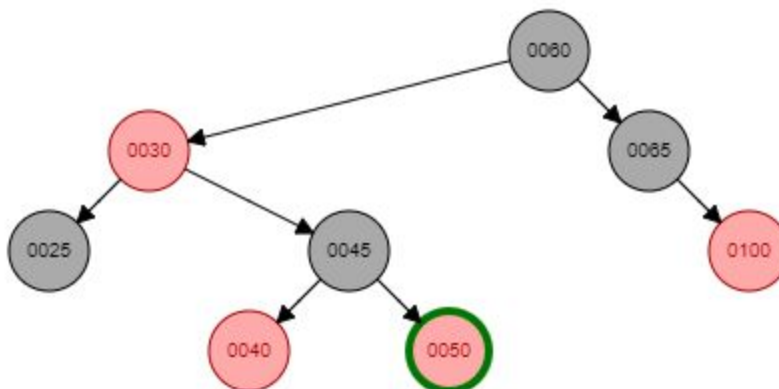
*Because there is no left side, set the left side of the parent of node 20 (node 30) to node 25 and change color for node 25, then delete node 20.

13. Delete: 60

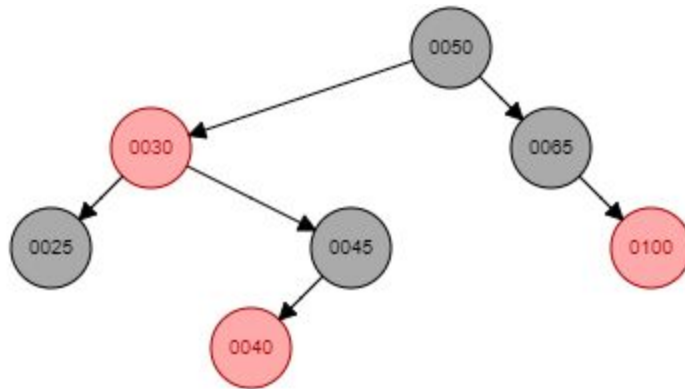
*traverse to find 60



*found 60, but 60 has children



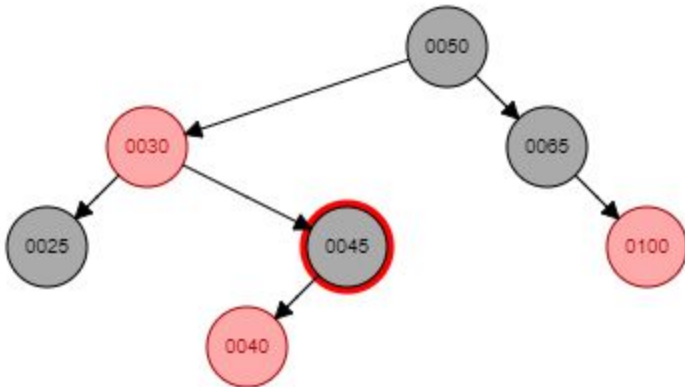
*traverse the left side, because 30 has a child, traverse right side until found a leaf.
Found 50



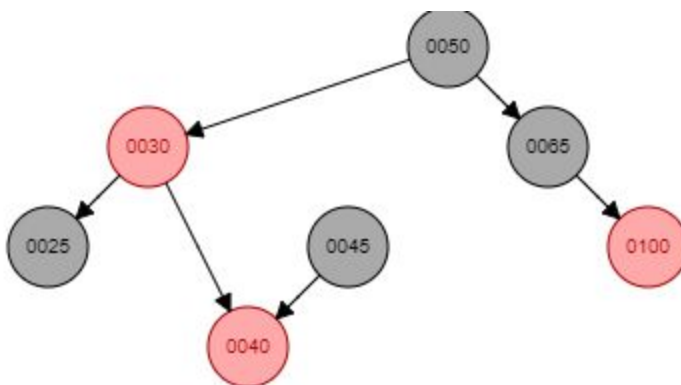
*50 will replace 60, then we just delete node 50.

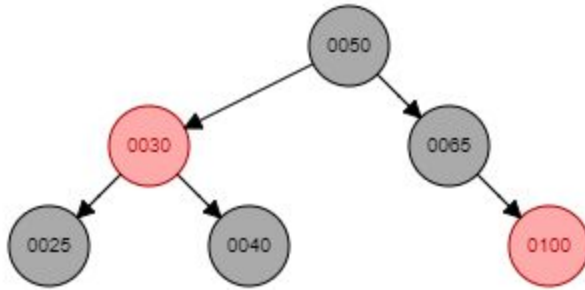
14. Delete: 45

*traverse to find 45



*found 45, but 45 has children

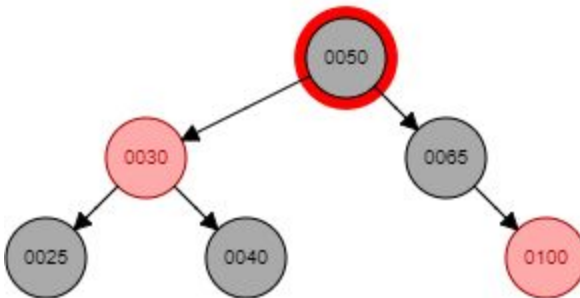




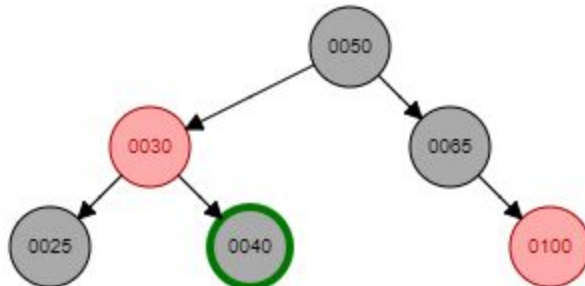
*traverse the left side, because 40 has no child, 40 will replace 45, then we delete node 50

15. Delete: 50

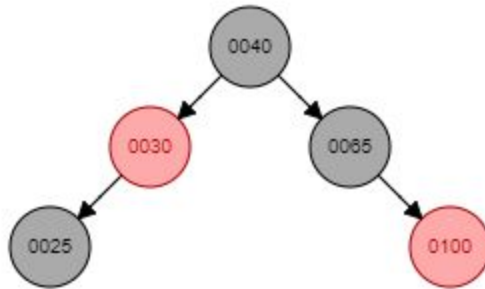
*traverse to find 50



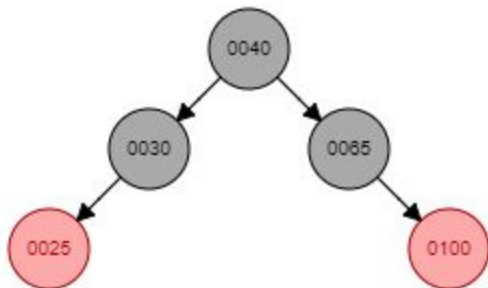
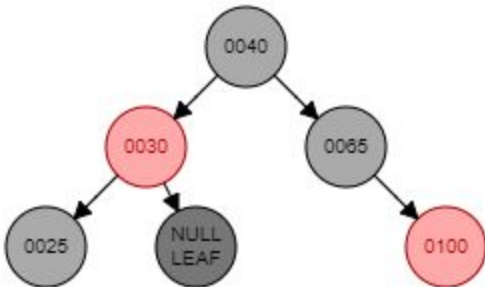
*found 50, but 50 has children



*traverse the left side, because 30 has a child, traverse right side until found a leaf.
Found 40



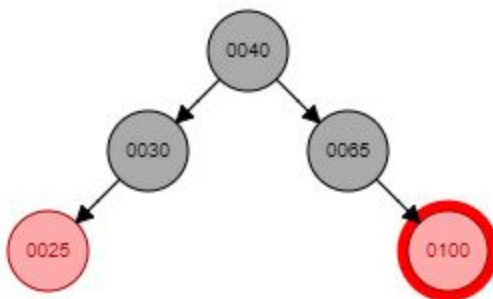
*40 will replace 50, then we delete node 40



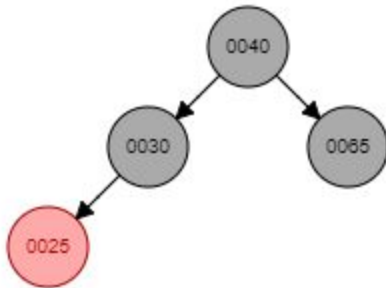
*coloring the child of the deleted node black (null leaf), but because it has black sibling and black uncle, push back the black color, and the child of 30 becomes red.

16. Delete: 100

*traverse to find 100



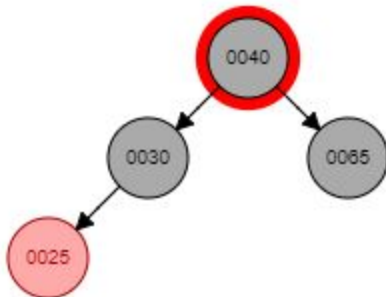
*found 100, and 100 doesn't have any child



*just delete node 100

17. Delete: 40

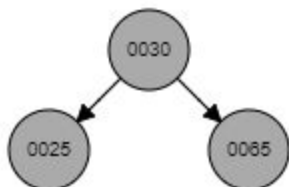
*traverse to find 40



*found 40, but 40 has a child



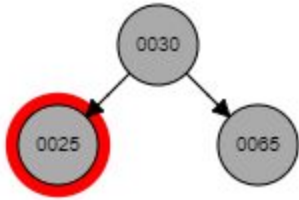
*go to left side, because there is no right side of 30, we replace 40 with 30 and connect the left side with its original child, then remove node 30



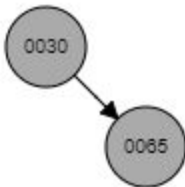
*color the deleted node black

18. Delete: 25

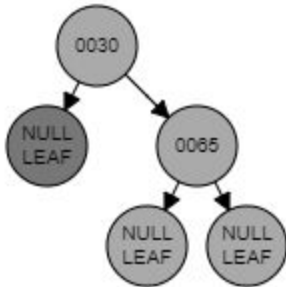
*traverse to find 25



*found 25, and 25 doesn't have any child



*just delete node 25



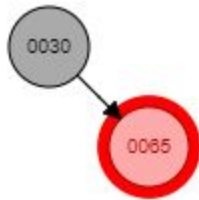
*coloring the left side (null) black, but it will have black sibling and black nephews



*push black up

19. Delete: 65

*traverse to find 65



*found 65, and 65 doesn't have any child



*just delete node 65

20. Delete: 30

*traverse to find 30



*found 30, and 30 doesn't have any child

*just delete node 30