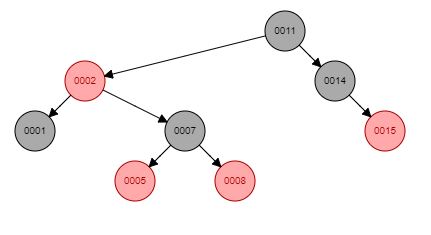
1. Inserting a node, re-balancing the tree, and then deleting the inserted node does not necessarily result in the original tree. Consider this initial tree:



Upon inserting a three, which would be added as the left child of five, the result would be a Case 1 violation—a red node with a red uncle, eight. This would be resolved by recoloring the grandparent, seven, to red, and its children, five and eight, to black. Now seven and two are both red and seven’s uncle, fourteen, is black, this has resulted in a Case 2 violation. Because its grandparent, the root, cannot be recolored without violating Property 5—which is that the root must be black—the solution is rotation. Because seven is the target node and it is a right child, the rotation will be leftward. This rotation is done about two, because it is seven’s parent. This results in a violation of Property 4—if a node is red, then both of its children must be black—because now both seven and its child, two, are red. Because the grandparent of two is the root and cannot be recolored, further rotation is necessary. The target node shifts upward from two to its parent, seven, which is a left child, which necessitates rightward rotation. Seven becomes the root, which means that its color must change to black. Because its right child, eleven,