Exercise 1 Simple training exercises

1. Draw the red-black tree that results after TREE-INSERT (ie. the insert procedure for normal binary search trees) is called on the tree in Figure 13.1 with key 36. If the inserted node is colored red, is the resulting tree a red-black tree? What if it is colored black? (CLRS 13.1-2)
2. In Figure 13.3 (after the rotation), perform a LEFT-ROTATE on the node with key 11, then a RIGHT-ROTATE on the node with key 14 and then a LEFT-ROTATE on the node with key 7.
3. Show the red-black trees that result after successively inserting the keys 41, 38, 31, 12, 19, 8 into an initially empty red-black tree. (CLRS 13.3-2)
4. Suppose that the black-height of each of the subtrees α, β, γ, δ, ϵ in Figures 13.5 and 13.6 is k. Label each node each figure with its black-height to verify that the indicated transformation preserves property 5. (CLRS 13.3-3)

Exercise 2 Fun creative exercises!

1. Line 16 of RB-INSERT sets the color of the newly inserted node z to red. If instead z's color were set to black, then property 4 of a red-black tree would not be violated. Why not set z's color to black? (CLRS 13.3-1)
2. Consider a red-black tree formed by inserting n nodes with RB-INSERT. Argue that if $n > 1$, the tree has at least one red node. (CLRS 13.3-5)

The next three questions kind of build on each other and constitutes a fun chain of reasoning!

3. Argue that in every n-node binary search tree, there are exactly $n - 1$ possible rotations. (CLRS 13.2-2)
4. Shoe that at most $n-1$ right rotations suffice to transform any n-node BST into a right-going chain.
5. Show that any arbitrary n-node binary search tree can be transformed into any other arbitrary n-node binary search tree using $O(n)$ rotations. (CLRS 13.2-4)