Assignment 4

Grades: each assignment is 5% in the final score and there is a 2% bonus in each assignment for bonus problems. We use the following scale for the simplicity of grading: the total score is 50 (additional 20 for bonus problems) for each assignment.

Problem 1 [Full Score: 10]. R.3-5 and R.3-6. (on page 212)

Problem 2 [Full Score: 10]. Find an example of an AVL tree where deleting a single node will require at least 2 restructure operations in order to restore the height balance property. Draw the tree both before deleting the node and after deleting the node rebalancing. Your answer should consist of two trees, and both must satisfy the height balance property.

Problem 3 [Full Score: 10]. R-3.11 part (a). (on page 212)

Problem 4 [Full Score: 10]. R-3.12. (on page 212) Your answer should consists of one (2,4) tree and four red-black trees that correspond to that (2,4) tree.

Problem 5 [Full Score: 10]. R-3.14: part(a) and part (b). (on page 213)

Problem 6 [Bonus Score: 20]. Consider the rotations in the insertion and the removal of AVL trees. Let \( w \) be the position to insert or to remove one node. Let \( z \) be the first unbalanced node along the path from \( w \) to the root. Let \( y \) be \( z \)'s child with larger height and \( x \) be \( y \)'s child with larger height. Consider the case where the inorder order of \( x, y, z \) is \( y, x, z \) (the double-rotation case). Let \( h_0, h_1 \) be the height function of nodes before and after the update respectively. Prove the following:

(1) [10] **Insertion:** prove that \( x, y, z \) are balanced after the update. Moreover, prove that \( h_1(x) = h_0(z) \).

(2) [10] **Removal:** prove that \( x, y, z \) are balanced after the update.

HINT: please review insertion and removal operations of AVL trees and the note on the correctness of the single-rotation case first. Note the difference between insertion and removal and carefully argue about the properties of \( h_0, h_1 \) for each case.