Ve281 Data Structures and Algorithms
Written Assignment Six

This assignment is announced on Nov. 20th, 2018. It is due by 5:40 pm on Nov. 30th, 2018. The assignment consists of four problems.

1. (30%) AVL tree

   (a) (6%) Suppose that we insert a sequence of keys 2, 1, 3, 7, 9 into an initially empty AVL tree. Draw the resulting AVL tree.

   (b) (6%) Suppose that we further insert key 6 into the AVL tree you get in Problem (1a). Draw the resulting AVL tree.

   (c) (6%) Suppose that we further insert keys 4, 5 into the AVL tree you get in Problem (1b). Draw the resulting AVL tree.

   (d) (6%) Suppose that we further insert keys 8, 10, 11, into the AVL tree you get in Problem (1c). Draw the resulting AVL tree.

   (e) (6%) Write down the balance factor for each node in the final AVL tree you get in Problem (1d).

2. (30%) Red-black tree

   (a) (6%) Suppose that we insert a sequence of keys 9, 3, 1 into an initially empty red-black tree. Draw the resulting red-black tree.

   (b) (4%) Suppose that we further insert key 6 into the red-black tree you get in Problem (2a). Draw the resulting red-black tree.

   (c) (6%) Suppose that we further insert keys 2, 8 into the red-black tree you get in Problem (2b). Draw the resulting red-black tree.

   (d) (4%) Suppose that we further insert key 7 into the red-black tree you get in Problem (2c). Draw the resulting red-black tree.

   (e) (10%) Suppose that we further insert keys 4, 5 into the red-black tree you get in Problem (2d). Draw the resulting red-black tree.

   When you draw the red-black tree, please indicate the color of each node in the tree. For example, you can color each node or put a letter b/r near each node.

3. (20%) 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. (Hint: First show that at most $n-1$ right rotations suffice to transform the tree into a right-skewed binary search tree.)

4. (20%) Suppose that an AVL tree insertion breaks the AVL balance condition. Suppose node P is the first node that has a balance condition violation in the insertion access path from the leaf. Assume the key is inserted into the left subtree of P and the left child of P is node A. Prove the following claims:

   (a) (8%) Before insertion, the balance factor of node P is 1. After insertion and before applying rotation to fix the violation, the balance factor of node P is 2.

   (b) (12%) Before insertion, the balance factor of node A is 0. After insertion and before applying rotation to fix the violation, the balance factor of node A cannot be 0.