Homework 3
Fundamental Algorithms, Spring 2008, Professor Yap

Due: Wed Mar 5, in class.

INSTRUCTIONS:

- Please read questions carefully. When in doubt, please ask.
- Please write succinctly, to the point.
- You may post general questions to the homework discussion forum in class website. Also, bring your questions to recitation on Monday.

---

1. (16 Points)
This is Exercise 2.1 (p.6) in Lecture III (please download the version dated Feb 27). Consider the dictionary ADT.
(a) Describe algorithms to implement this ADT when the concrete data structures are linked lists. Try to be brief and give algorithms at the high level that we use in our lecture notes.
(b) Analyze the worst complexity of your algorithms in (a). NOTE: complexity is be analyzed to Θ-order.
(a') Describe algorithms to implement this ADT when the concrete data structures are arrays instead of linked lists.
(b') Analyze the complexity of your algorithms in (a').

2. (12 Points)
Consider the priority queue ADT.
(a) Describe algorithms to implement this ADT when the concrete data structures are AVL trees. You may assume the standard AVL algorithms.
(b) Analyze the complexity of your algorithms in (a).

3. (0 Points – do not hand in!)
Do Exercise 3.1 in Lecture III (delete key 10 from the BST of Figure 3(a), using rotation-based and standard deletion algorithms).

4. (10 Points)
This is Exercise 3.11 (page 17) in Lecture III. Prove Lemma 2 (p. 14)that there is a unique way to order the nodes of a binary tree T that is consistent with any binary search tree based on T. HINT: remember the fundamental rule about binary trees!

5. (9 Points)
Exercise 3.5 (p.16) in Lecture III. Let T be the binary search tree in Figure 3(a) in Lecture III. HINT: although we only require that you show the trees at the end of the operations, we recommend that you show selected intermediate stages. This way, we can give you partial credits in case you make mistakes!
(a) Perform the operation Display T and T' after the split.
(b) Now perform tree after the operation in (a). Display the tree after insertion.
(c) Finally, perform tree after the insert in (b) and T' is the tree after the split in (a).

6. (12 Points)
This is Exercise 3.19 (page 18) in Lecture III. Suppose we allow allow duplicate keys. Under the binary search tree property (equation (1) in p.7), we see that all the keys with the same value must lie in consecutive nodes of some "right-path chain".
(a) Show how to modify lookup on key K so that we list all the items whose key is K.
(b) Discuss how this property can be preserved during rotation, insertion, deletion.
(c) Discuss the effect of duplicate keys on the complexity of rotation, insertion, deletion. Suggest ways to improve the complexity.

7. (12 Points)
   This is Exercise 4.2 (p. 21) in Lecture III on tree traversals.
   (a) Let the in-order and pre-order traversal of a binary tree T with 10 nodes be $(a, b, c, d, e, f, g, h, i, j)$ and $(f, d, b, a, c, e, h, g, j, i)$, respectively. Draw the tree T.
   (b) Prove that if we have the pre-order and in-order listing of the nodes in a binary tree, we can reconstruct the tree.
   (c) Consider the other two possibilities: (c.1) pre-order and post-order, and (c.2) in-order and post-order. State in each case whether or not they have the same reconstruction property as in (b). If so, prove it. If not, show a counter example.
   (d) Redo part(c) for full binary trees. Recall that in a full binary tree, each node either has no children or 2 children.

8. (5 Points)
   This is Exercise 6.3, Lecture III. What is the minimum number of nodes in an AVL tree of height 10?

9. (0 Points, do not hand in!)
   Exercise 6.6, p.32. Referring to Figure 17:
   (a) Find all the keys that we can delete so that the rebalancing phase requires two rebalancing acts.
   (b) Among the keys in part (a), which deletion has a double rotation among its rebalancing acts?
   (c) Please delete one such key, and draw the AVL tree after each of the rebalancing acts.

10. (6 Points)
    Exercise 6.8, p.32. Draw the AVL trees after you insert each of the following keys into an initially empty tree: $1, 2, 3, 4, 5, 6, 7, 8, 9$ and then $19, 18, 17, 16, 15, 14, 13, 12, 11$.

11. (10 Points)
    For the midterm, you only need to understand 2-3 trees and not the general (a, b)-trees.
    (a) Show the result of inserting the keys $1, 3, 4, 20, 22, 24$ (in this order) into the tree in Figure 21 (p. 36). As usual, show intermediate results.
    (b) Show the result of deleting the keys 8 and then 10 from tree in Figure 21.