CSE 331 Algorithms and Data Structures
Homework 3

solution

1. Do Weiss textbook problem 4.5 (6 points)
   Answer: For a binary tree of height h, the maximum nodes can be obtained when each internal node has two branches. Thus, for a tree with height h, the total number of nodes is $1 + 2^1 + 2^2 + 2^3 + \ldots + 2^h$. This is a geometric series and you should be able to solve it easily. Alternatively, you can prove this theorem by induction. The theorem is trivially true for $h = 0$. Assume true for $h = 1, 2, \ldots, k$. A tree of height $k + 1$ can have two subtrees of height at most k. These can have at most $2^{k+1} - 1$ nodes each by the induction hypothesis. These $2^{k+2} - 2$ nodes plus the root prove the theorem for height $k + 1$ and hence for all heights.

2. Do Weiss textbook problem 4.9. For part a, show the tree after each insertion. (a. 8 points; b. 2 points)
   see the scanned file hw3-problem2.pdf

3. Do Weiss textbook problem 4.19. Show the tree after each insertion and rotation. (11 points)
   Answer: you need to perform 1 single rotation after inserting 9. 1 double rotation after inserting 3. 1 double rotation after inserting 6. The final tree has root 4, 4.leftchild=2, 4.rightchild=6, 2.leftchild=1, 2.rightchild=3, 6.leftchild=5, 6.rightchild=9, 9.leftchild=7.

4. Problem 4.48 (5 points).

   Answer: Add a data member to each node indicating the size of the tree it roots. This allows computation of its inorder traversal number.

5. What is the minimum number of nodes in an AVL tree of height 0? What are the minimum number of nodes in AVL trees of heights 1,

2, and 3? Let N(h) be the minimum number of nodes in an AVL tree with height h. Define a recursive function for N(h). (6 points)
Answer: N(0)=1(1 point); N(1)=2 (1 point); N(2)=4 (1 point); N(3)=7 (1 point). N(h)=N(h-1)+N(h-2)+1 (2 points);

6. Professor W thinks he has discovered a remarkable property of binary search trees. Suppose that the search for key k in a binary search tree ends up in a leaf. Consider three sets: A, the keys to the left of the search path; B, the keys on the search path; and C, the keys to the right of the search path. Professor W claims that any three keys $a \in A$, $b \in B$, and $c \in C$ must satisfy $a \leq b \leq c$. Give a smallest possible counterexample to the professor's claim. (4 points)

   Answer: example. root 4, 4.rightchild=6. 6.leftchild=5. 6.rightchild=9. key k=9 B=$\{4, 6, 9\}$, A=$\{5\}$, but 5>4.

7. Given a node in a BST, it is sometimes important to be able to find its successor in the sorted order determined by an inorder tree walk. If all keys are distinct, the successor of a node x is the node with the smallest key greater than key[x]. The structure of a BST allows us to determine the successor of a node easily. Describe a procedure that returns the successor of a node x in a binary search tree if it exists, and NIL if x has the largest key in the tree. (Hint: in order to consider all cases, find each node's successor in the tree for Exercise 4.27. Then generalize your observations.) (8 points)
   Answer: two cases. 1) node x has a right child. In this case, the minimum key in x's right subtree is x's successor. (3 points) case 2): node x has no right child. Follow the path from x to the root (unique path), the first node with left child is x's successor. When such node does not exist, x has no successor. (5 points) If you consider the case that x has the maximum key value in the tree, but not the general case that x has no right child, you get 2 points.