CSC 520
Homework 3
Due: September 29, 2010

1. Write an algorithm to count the number of leaf nodes in a binary tree T. What is the time complexity of your algorithm?

2. Write an algorithm to find the height of a binary tree T. What is the time complexity of your algorithm?

3. Write an algorithm to print the INFO fields of the nodes in a binary tree T level by level. Within levels, nodes are to be listed left to right.

4. Write an algorithm SWAPTREE(T) which takes a binary tree and swaps the left and right children of each node. What is the time complexity of your algorithm?

5. Given the Child-Sibling representation of a general tree T, write an algorithm to find the height of the original (general) tree (not the Child-Sibling binary tree).

6. Write an algorithm to delete a node from a binary search tree. Your algorithm will manipulate the left and right links of nodes, but the node objects (for example, the INFO field) should not be copied.

7. Suppose you are given a binary search tree in which each node contains an additional field, *size*, which contains the number of keys in the subtree rooted at this node. (Thus the *size* field of a leaf is 1 and the *size* field of the root is the total number of nodes in the tree, n.) Using this *size* field, give pseudocode for a procedure *findKth(t, k)* which returns the k-th smallest element in the search tree t. (You may assume that $1 <= k <= n$)