# CS3323 Fall 2006 Assignment 4

Due Monday, Nov. 13, by 5pm.

- Assignments should be handed in by placing them in the CS3323 bin on E level of Gillin Hall.

1. Design algorithms for performing the following operations on a binary tree $T$ of size $n$, and analyze their worst-case running time. Your algorithms should avoid performing traversals of the entire tree.

   (a) preorderNext($v$): return the node visited after node $v$ in a preorder traversal of $T$.

   (b) inorderNext($v$): return the node visited after node $v$ in an inorder traversal of $T$.

2. Let $T$ be a binary tree with $n$ nodes. It is realized with an implementation of the Binary Tree ADT that has $O(1)$ running time for all methods except *positions*() and *elements*(), which have $O(n)$ running time. Give an $O(n)$ time algorithm that uses the methods of the Binary Tree ADT to visit the nodes of $T$ by ***level order traversal***. ***Level order traversal*** visits the nodes in order of increasing depth, visiting the nodes at a given depth from left-to-right. Assume the existence of an $O(1)$ time visit($v$) method (it should get called once on each vertex of $T$ during the execution of your algorithm).

3. (a) Insert into an initially empty binary search tree items with the following keys (in this order): 30, 40, 23, 58, 48, 26, 11, 13. Draw the tree after each insertion.

   (b) Remove from the binary search tree built from (a) the following keys (in this order): 13, 40, 23. Draw the tree after each removal.

4. Let $T$ be a binary search tree, and let $x$ be a key. Give an efficient algorithm for finding the smallest key $y$ in $T$ such that $y > x$. Note that $x$ may or may not be in $T$. Explain why your algorithm has the running time it does.

5. Let $T$ be a heap storing $n$ keys. Give an efficient algorithm for reporting all the keys in $T$ that are smaller than or equal to a given query key $x$ (which is not necessarily in $T$). Note that the keys do not need to be reported in sorted order. Your algorithm should run in $O(k)$ time, where $k$ is the number of keys reported.

6. Illustrate the execution of the heap-sort algorithm on the following input sequence: $(2, 5, 16, 4, 10, 23, 39, 18, 26, 15)$. Show the contents of both the heap and the sequence at each step of the algorithm.