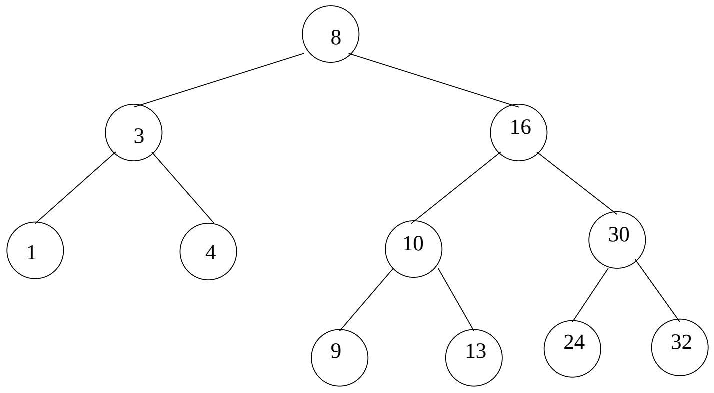
CSE 326
Winter 2008
Assignment 4
Due 2/8/08

For all algorithm and data structure design problems please provide elegant pseudocode and an adequate explanation of your methods. If is often helpful to include small examples demonstrating the method. Put your name at the top of each sheet of paper that you turn in.

1. Consider the task of printing a range of values that are stored in a binary search tree. For example, for the following tree:

   a call to PrintRange(root, 4, 24) would print 4 8 9 10 13 16 24. Present pseudocode for an efficient recursive implementation of PrintRange(root: node pointer, low: integer, high: integer) You may assume that the values low and high actually appear in the tree.

   Analyze your algorithm, and prove that if the tree is complete (perfectly balanced) it runs in time $O(k + \log n)$ where n is the number of nodes in the tree, and k is the number of values printed out.

2. The main operations for binary trees are insert, delete, and find. Suppose we would like to add rank which given an integer k returns the node in the tree which has the k-th largest key. In order to implement rank, it is very handy to add a new field to each node which contains the number of descendents of the node in the tree. A node then has fields `[key: integer; count: integer; left,right: node pointer]`.

   (a) For the tree in problem 1 calculate the count for each node. The root has count 11.

(b) Design an algorithm and present the pseudocode for the function `rank(T: node pointer; k: integer): node pointer` which returns a pointer to the node whose key is the k-th largest. If k is larger than the number of nodes in the tree then return null.

(c) Design a recursive algorithm and present the pseudocode for insert which correctly maintains the count field. The algorithm should run in time bounded by a constant times the height of the tree.