Problem Set 3-b

Problem 1 For each node v in a tree T, let pre(v) be the rank of v in the preorder traversal of T, post(v) the rank of in the postorder traversal of T, depth(v) the depth of v, and desc(v) the number of descendants of v, not counting v itself. Derive a formula defining post(v) in terms of desc(v), depth(v), and pre(v), for each node v in T.

Problem 2 Give an O(n)-time algorithm for computing the depths of all the nodes of a tree T, where n is the number of nodes of T.

Problem 3 Describe, in pseudo code, a nonrecursive algorithm for performing the preorder traversal on a binary tree in linear time.

Problem 4 Describe, in pseudo code, a nonrecursive algorithm for performing the inorder traversal on a binary tree in linear time.

Problem 5 Describe, in pseudo code, a nonrecursive algorithm for performing the postorder traversal on a binary tree in linear time.

Problem 6 In the inorder traversal on a binary tree, a node v_i is the immediate inorder predecessor of a node v_j, if v_j is visited immediately after v_i in the inorder traversal. Describe a linear time algorithm for finding the immediate inorder predecessor of a node.

Problem 7 Let T be a binary tree. Define a **Roman node** to be a node v in T, such that the number of descendants in v's left subtree differs from the number of nodes in v's right subtree by at most 5. Describe a linear-time algorithm for finding each node v of T, such that v is not a Roman node, but all of v's descendants are Roman nodes.

Problem 8 Let T be a binary tree with n nodes. Define the lowest common ancestor (LCA) of two nodes v and w as the lowest node in T that has both v and w as descendants (where we allow a node to be a descendant of itself). Given two nodes v and w, describe an efficient algorithm for finding the LCA of v and w. What is the running time of your algorithm?