1. Label the nodes of a complete binary tree of height 3 with numbers 1 to 15, such that on performing the post order traversal the labels appear in strictly increasing order.

2. Write a recursive and a non-recursive algorithm to calculate the height of an ordered tree. Compare the running times of both the algorithms.

3. Given a binary tree representing an arithmetic expression. Write an algorithm (1) to evaluate the expression, (2) print the expression in fully-parenthesised form.

4. Write a non-recursive preorder traversal algorithm for a proper binary tree.

5. Given a node $v$ of a proper binary tree, write an algorithm to find the (1) inorder successor, (2) postorder successor, and (3) preorder successor of the node $v$.

6. Let $T$ be a binary tree with $n$ nodes such that all the external nodes have the same depth. Let $D_e$ be the sum of the depths of all the external nodes, and let $D_i$ be the sum of the depths of all the internal nodes. Establish a relation between $D_i, D_e$ and $n$ and prove it.

7. Describe how to implement a Queue using two Stacks. Find the amortized running time for dequeue and enqueue operations.
8. Describe how to implement a Stack using two Queues. What is the running time of the \textit{push} and \textit{pop} operations.

9. Let $T$ be a tree with $n$ nodes. Define the lowest common ancestor (LCA) between the two nodes $v$ and $w$ as the lowest node in $T$ that has both $v$ and $w$ as descendents. Given $v$ and $w$, describe an efficient algorithm to find the LCA of $v$ and $w$. What is the running time.

10. The \textit{Balance Factor} of an internal node $v$ of a binary tree is the difference between the heights of the right and left subtrees of $v$. Write a recursive algorithm to print the balance factor of all the nodes of a given binary tree.