SHORT ASSIGNMENT NUMBER 2

1. Is it true that the leaf nodes appear in the same relative order for the preorder, inorder, and postorder traversals of a binary tree?

2. Calculate the total number of leaf nodes in a tree having a_i nodes of degree i ($1 \leq i \leq n$).

3. Suppose that you are given the inorder and preorder traversals of a binary tree. Can you reconstruct the binary tree?

4. Describe the set of binary trees whose preorder and inorder traversals appear in the same order.

5. Define an *extended binary tree* to be a binary tree such that each of its non-empty nodes has two sons. This means that every leaf node has a null left son and a null right son. Nodes in the original tree that only have a left son are assigned a null right son in the extended binary tree. Nodes in the original tree that only have a right son are assigned a null left son in the extended binary tree. Prove that a binary tree of n nodes has $n + 1$ null sons.

6. Consider a binary tree that is threaded in inorder. Note that some nodes have several threads pointing at them. What is the maximum number of threads that can point at a node? Describe the binary trees that can cause these situations to arise.

7. Describe how to find the inorder predecessor of an arbitrary node P in a binary tree that is threaded in inorder.

8. Devise an algorithm to traverse a binary tree in inorder that does not make use of a stack or threads. You may temporarily change the values of the pointer fields during this process. However, at the end of the algorithm all pointer fields are to have the values they had prior to the invocation of the algorithm. You may make use of an additional one-bit `FLAG` field in each node for auxiliary storage.

The following exercises from the "Notes on Data Structures" are optional. You are urged to look at them for practice.

6, 7, 8, 32, 33, 36, 37, 10, 11, 12, 13, 14, 15, 16, 17, 42, 43, 18, 19, 20, 21, 44, 45