ASSIGNMENT NUMBER 2

You are to turn in solutions to exercises 1, 5, 6, 9, 12, 13, 18, and 31. The rest are optional. You are urged to look at them for practice.

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. Give an order for visiting the nodes of a binary tree that yields the reverse of preorder.

3. Give an order for visiting the nodes of a binary tree that yields the reverse of inorder.

4. Give an order for visiting the nodes of a binary tree that yields the reverse of postorder.

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

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

7. Repeat Exercise 6 for the inorder and postorder traversals of a binary tree.

8. Repeat Exercise 6 for the preorder and postorder traversals of a binary tree.

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

10. Describe the set of binary trees whose preorder and postorder traversals appear in the same order.

11. Describe the set of binary trees whose inorder and postorder traversals appear in the same order.

12. 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. We use the qualifier *external* to refer to all nodes in the extended binary tree that correspond to null sons in the original tree, and *internal* to refer to all nodes in the original binary tree. Prove that the extended binary tree corresponding to a binary tree of n nodes (i.e., internal nodes) has $n + 1$ null sons (i.e., external nodes).

13. 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.

14. Describe how to find the inorder predecessor of a nonleaf node P in an unthreaded binary tree without making use of `FATHER` pointers. Assume that P has a nonempty left son.

15. Describe how to find the inorder predecessor of an arbitrary node in an unthreaded binary tree by making use of `FATHER` pointers if necessary.

16. Describe how to find the inorder successor of a nonleaf node P in an unthreaded binary tree without making use of `FATHER` pointers. Assume that P has a nonempty right son.

17. Describe how to find the inorder successor of an arbitrary node in an unthreaded binary tree by making use of `FATHER` pointers if necessary.

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

19. Describe how to find the preorder predecessor of a nonleaf node P in an unthreaded binary tree without making use of `FATHER` pointers.

20. Describe how to find the preorder predecessor of an arbitrary node P in an unthreaded binary tree by making use of `FATHER` pointers if necessary.

21. Describe how to find the preorder successor of a nonleaf node P in an unthreaded binary tree without making use of `FATHER` pointers.

22. Describe how to find the preorder successor of an arbitrary node P in an unthreaded binary tree by making use of `FATHER` pointers if necessary.

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

24. Describe how to find the preorder successor of an arbitrary node P in a binary tree that is threaded in inorder.

25. Describe how to find the postorder predecessor of a nonleaf node P in an unthreaded binary tree without making use of `FATHER` pointers.

26. Describe how to find the postorder predecessor of an arbitrary node P in an unthreaded binary tree by making use of `FATHER` pointers if necessary.

27. Describe how to find the postorder successor of a nonleaf node P in an unthreaded binary tree without making use of `FATHER` pointers.

28. Describe how to find the postorder successor of an arbitrary node P in an unthreaded binary tree by making use of `FATHER` pointers if necessary.

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

30. Describe how to find the postorder successor of an arbitrary node P in binary tree that is threaded in inorder.

31. 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.