Tree Traversals

See Section 18.4 of Weiss.
We might create Binary Tree structures with the following Node class

class Tree <T> {
    T data;
    Tree<T> left;
    Tree<T> right;
    ....
Here is a picture of a Binary Tree with Integer values at each node:
There are three standard ways to iterate through the nodes of a tree:

A *preorder traversal* of a tree lists the root, then its left subtree, then its right subtree.

For this tree the preorder traversal is

10  17  34  23  19  45  12  63
A postorder traversal of a tree lists the left subtree, the right subtree and then the root.

For this example the postorder traversal is
34  17  45  12  19  63  23  10
Finally, the *inorder traversal* lists the left subtree, the root, and then the right subtree.

For this example the inorder traversal is
34 17 10 45 19 12 23 63
If all that you want to do is to print the data stored in a node in the order of one of these traversals, a simple recursion does the job. Here is the preorder traversal, assuming that empty trees are represented by null pointers.

```java
public void PrintPreorder(TreeNode t) {
    if (t != null) {
        System.out.println(t.data);
        PrintPreorder(t.left);
        PrintPreorder(t.right);
    }
}
```