5. Trees :: Binary Trees :: Traversals :: Depth First :: Postorder

A **postorder** traversal is one in which we traverse the left subtree of the root node first, then we traverse the right subtree of the root node, and then we visit the root node. For the example tree, the nodes would be visited in this order:

Note that our definition of postorder traversal is recursive in nature:

1. Perform a postorder traversal of the subtree rooted at the left child of the root node (if it exists).
2. Perform a postorder traversal of the subtree rooted at the right child of the root node (if it exists).
3. Visit the root node.

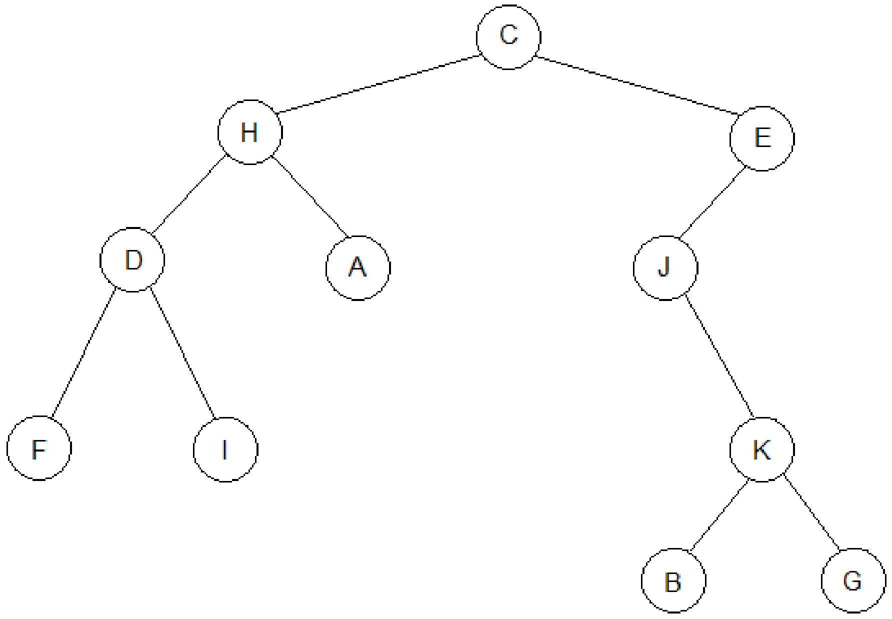
Here is the pseudocode:

```
-- Performs a postorder traversal of the subtree rooted at pRoot. pVisitor is an object which
-- implements a method named visit() which will be called as each Node is visited.
Method postorderTraversal(In: Node pRoot, In: pVisitor) Returns Nothing
    -- Perform a postorder traversal of the subtree rooted at the left child pRoot if it exists.
    If pRoot has a left child Then postorderTraversal(pRoot.leftChild, pVisitor)

    -- Perform a postorder traversal of the subtree rooted at the right child pRoot if it exists.
    If pRoot has a right child Then postorderTraversal(pRoot.rightChild, pVisitor)

    -- Visit the root node passing the data stored in the root node.
    pVisitor.visit(pRoot.data)
End Method postorderTraversal
```

5. Trees :: Binary Trees :: Traversals :: Depth First :: Postorder


C
H
E
D
A
J
F
I
K
B
G