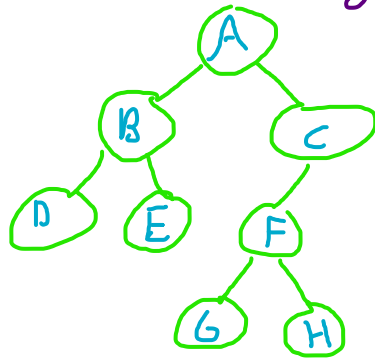


Level-Order Traversal:

Top to bottom. Left to right (technically doesn't matter)



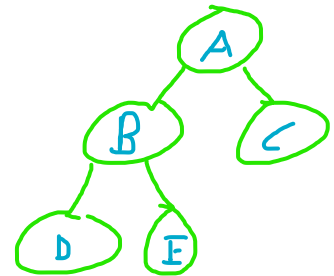
Order:
A, B, C, D, E, F, G, H

Depth-First Traversal:

Try to explore deep nodes before "backtracking."

Pre-Order:

```
public void preorder (Node node) {  
    System.out.print (node.value);  
    preorder (node.left);  
    preorder (node.right);  
}
```

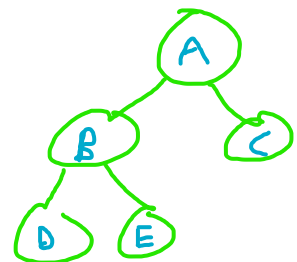


Order: A, B, D, E, C

In-Order:

```
public void inorder (Node node) {  
    → inorder (node.left);  
    System.out.print (node.value);  
    inorder (node.right);  
}
```

recursively
goes to left
bottom node first

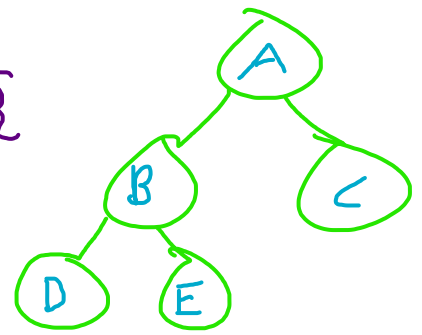


Order: D, B, E, A, C

What do we notice? We print (or "visit") nodes from leftmost to rightmost.

Post - Order

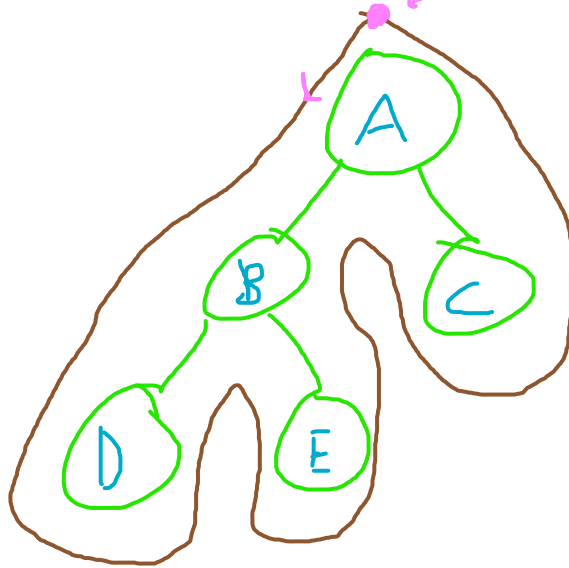
```
public void postOrder(Node node){  
    postOrder(node.left);  
    postOrder(node.right);  
    System.out.print(node.value);  
}
```



Order: D, E, B, C, A

"For EVERY node in the tree,
visit all nodes on the left, then
visit all nodes on the right, then
visit self"

Tricks:



preorder: A, B, D, E, C

inorder: D, B, E, A, C

postorder: D, E, B, C, A

① start here

② Trace the brown
path going left w/ finger

③ Whenever finger on
left of a node,
add node to preorder.

Whenever finger bottom
of a node, add node
to inorder.

Whenever finger on right
of a node, add node
to postorder.