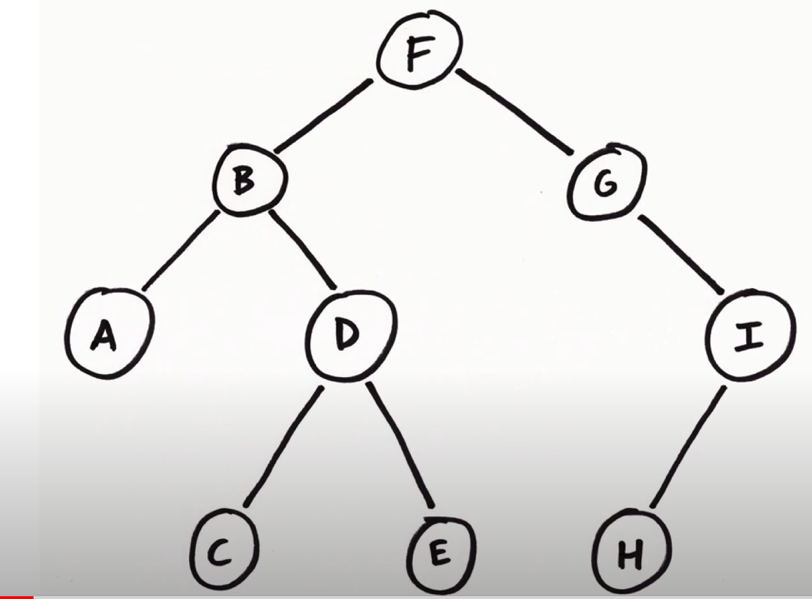
**Traverse a Binary Tree or Binary Search Tree**

This is a Binary Tree:



Tree Travesral – process of visiting (read/process data in node) each node in the tree exactly once in some order

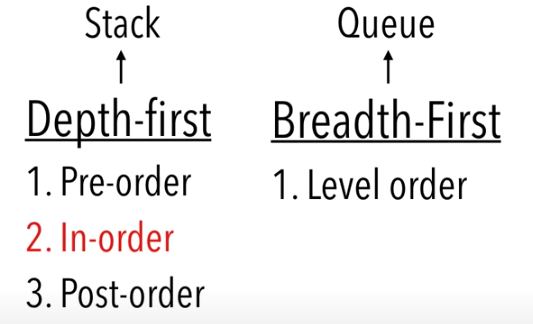
Trees can be traversed in several ways, but we always have the restriction that the left child is visited before the right child.

The main difference is where is the node to be visited.

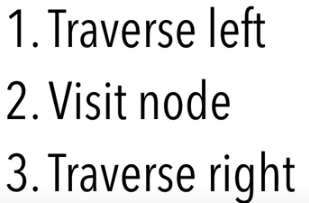
* If the node is visited **before** both children, we call it preorder. (Visit, Left, Right)
* If the node is visited **after** both children, we call it postorder. (Visit, Right, Node)
* If the node is visited in between both children, we call it inorder. (Visit, Node, Right)
* **Pre Order Traversal**: Visit, Left, Right.
* **In Order Traversal**: Left, Visit, Right.
* **Post Order Traversal**: Left, Right, Visit.
* **Level Order Traversal**, also known as Breadth-first search.

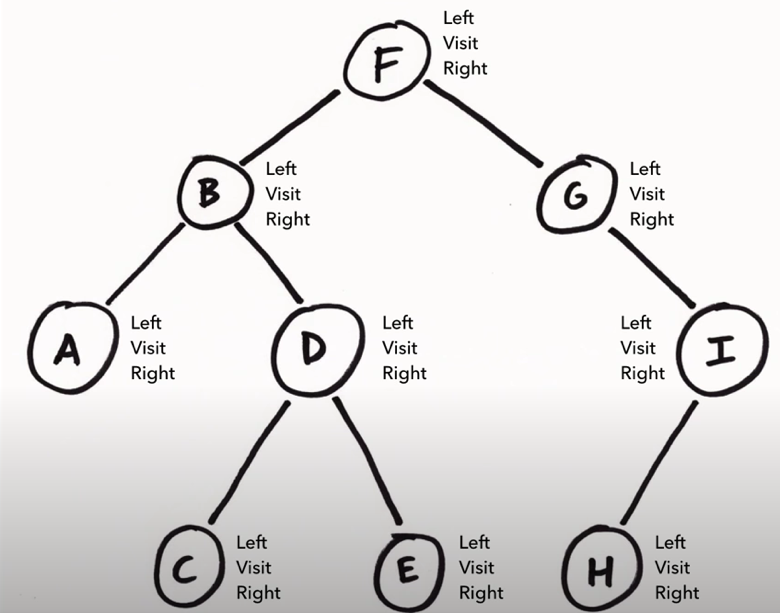
<https://www.youtube.com/watch?v=5dySuyZf9Qg&t=1s>

<https://medium.com/swlh/tree-traversal-algorithms-theory-and-practice-in-java-7e7d1fe9ed30>



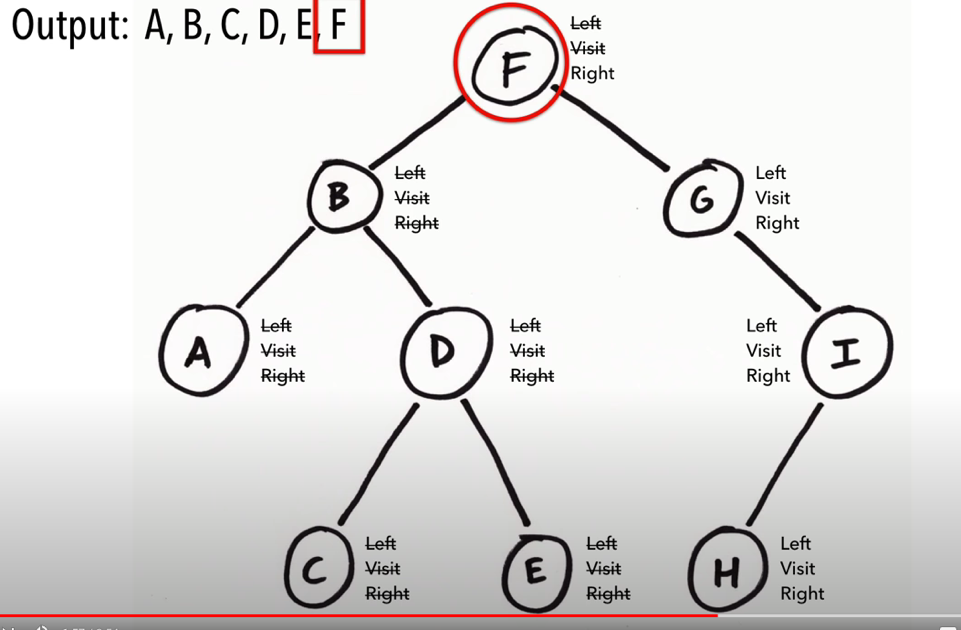
**In-Order**



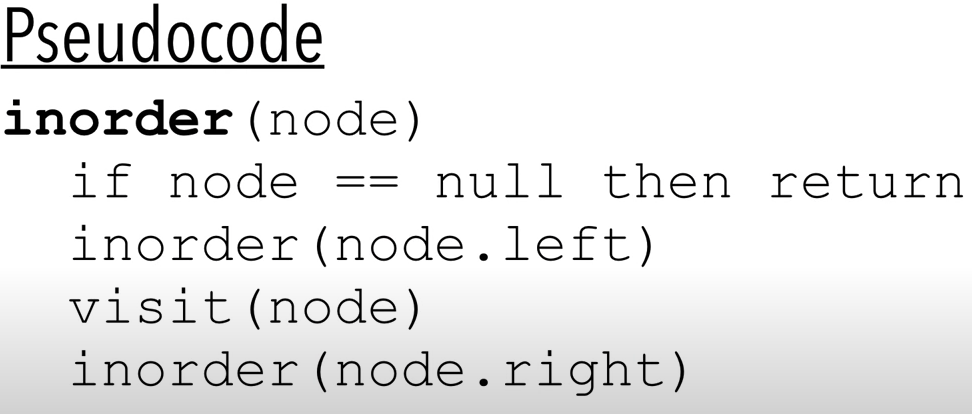


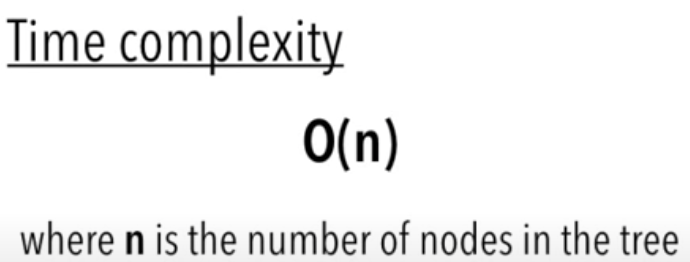
Start at top with F. Travesre left subtree to B (cross out left)-> Traverse left subtree to A (cross out left)-> Vist A (cross out visit) -> Travesre right subtree (no right so cross out right).

If a node doesn’t have children then return to its parents



A,B,C,D,E,F,G,H,I

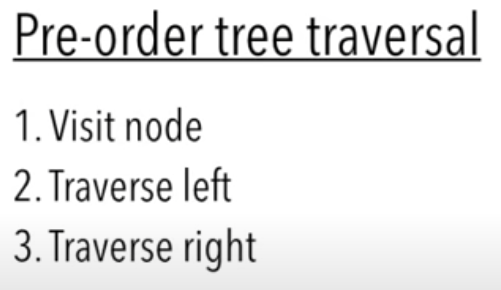


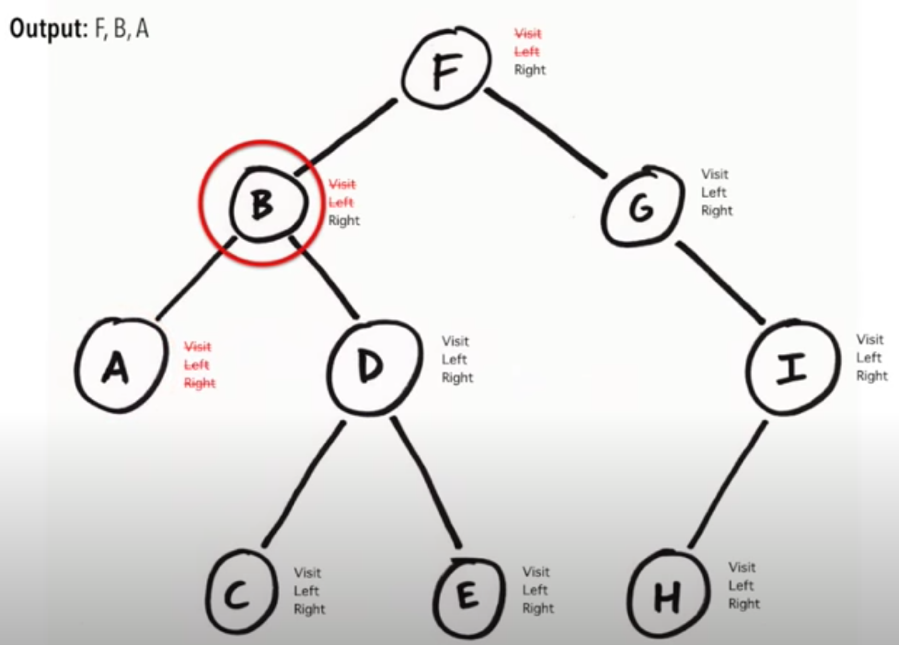


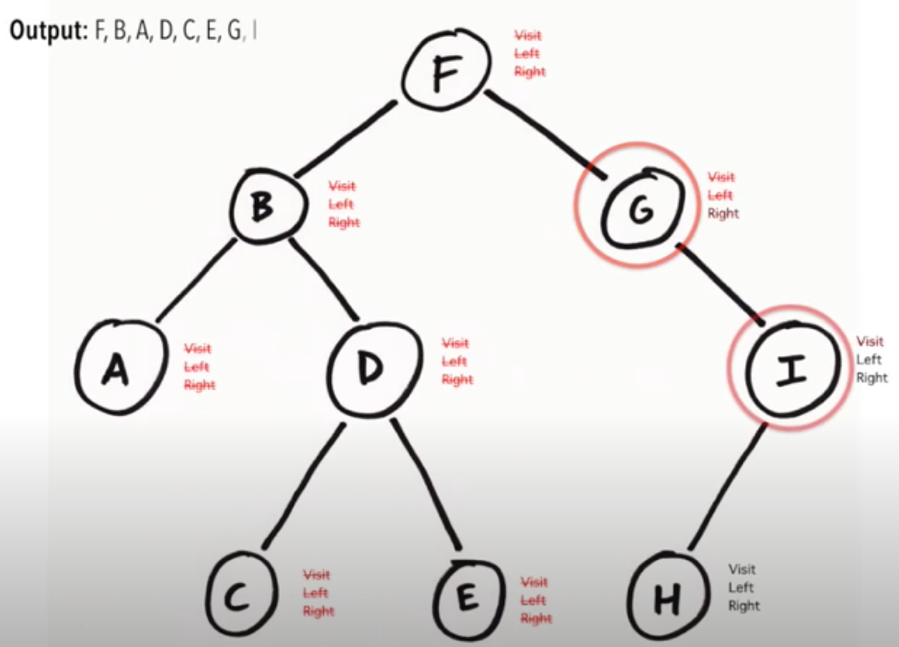
Complexity is o(n) because we call visit on each node exactly once.

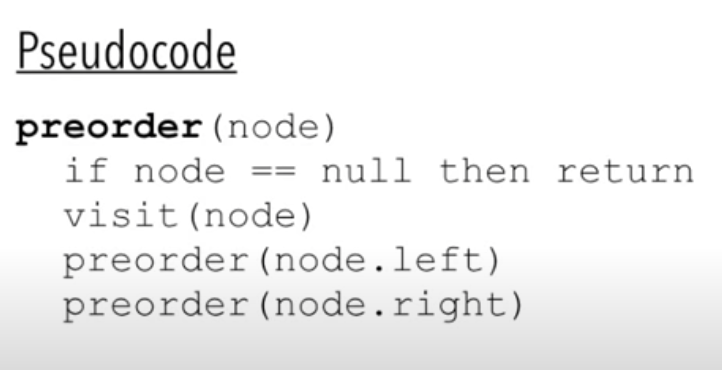


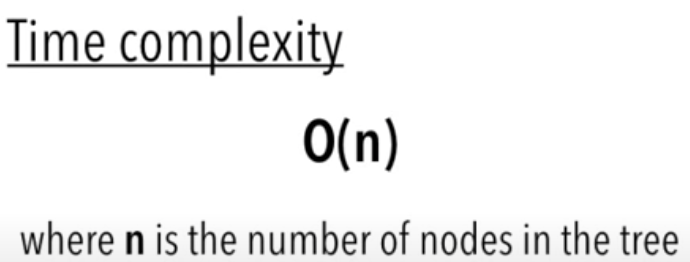
**Pre-Order**



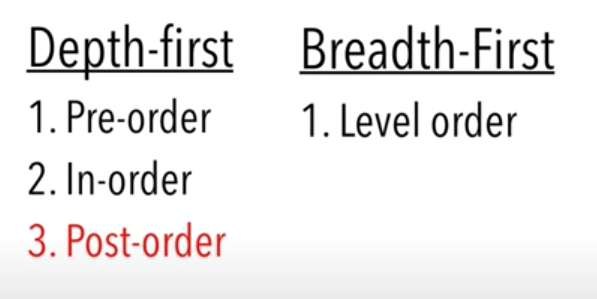




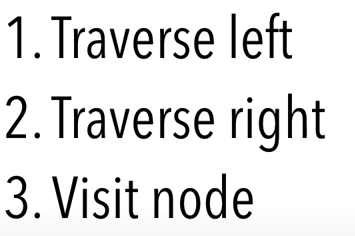


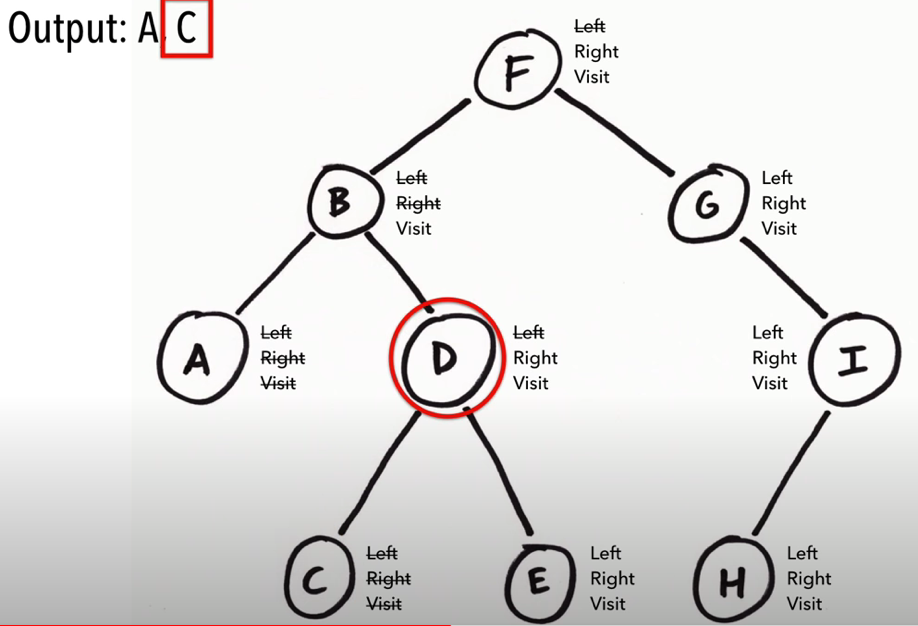


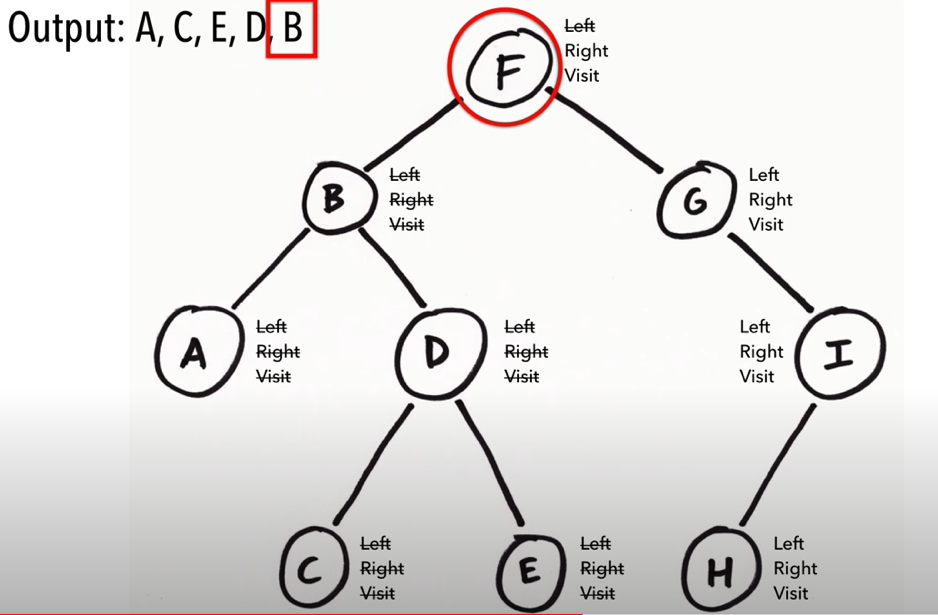
Complexity is o(n) because we call visit on each node exactly once.

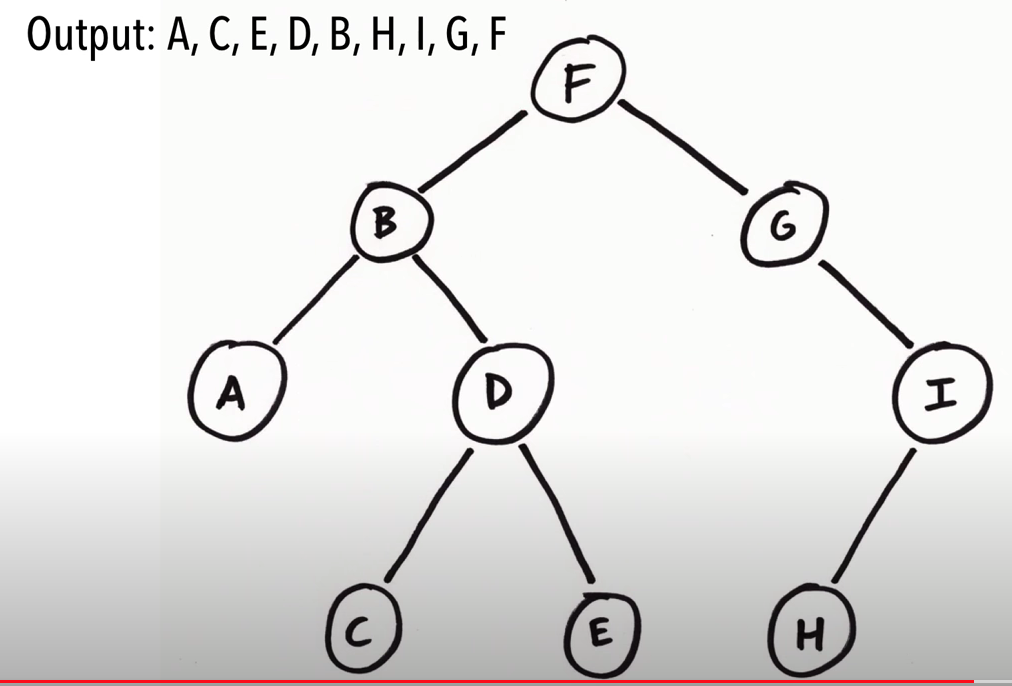


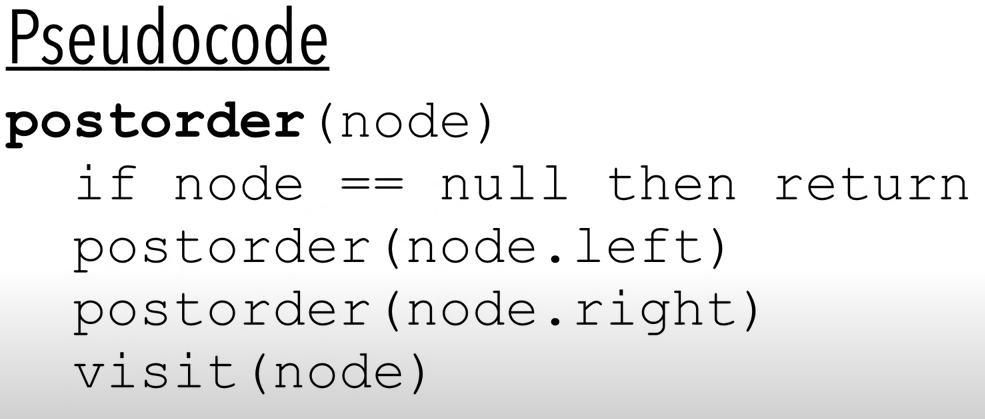
**Post-Order**

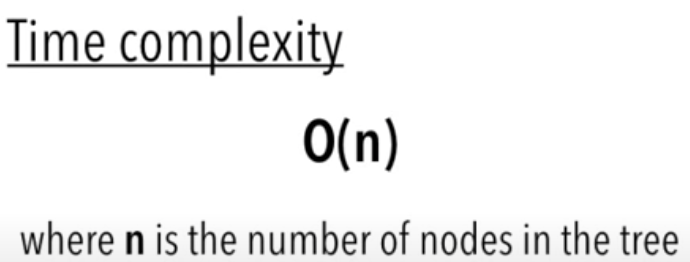




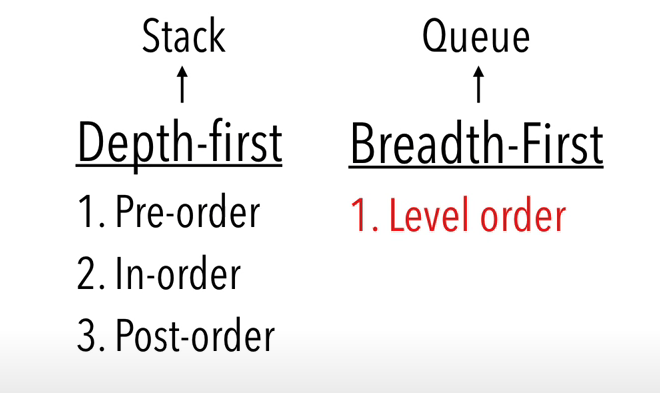








Complexity is o(n) because we call visit on each node exactly once.



**Level-Order**

<https://tutorialedge.net/artificial-intelligence/breadth-first-search-java/>

TODO – explain how level-order (dfs) actually works and is implemented with queue.