

Traversals of A Tree

- Visit each data item exactly once
- A traversal can pass through a node without visiting it at that moment.

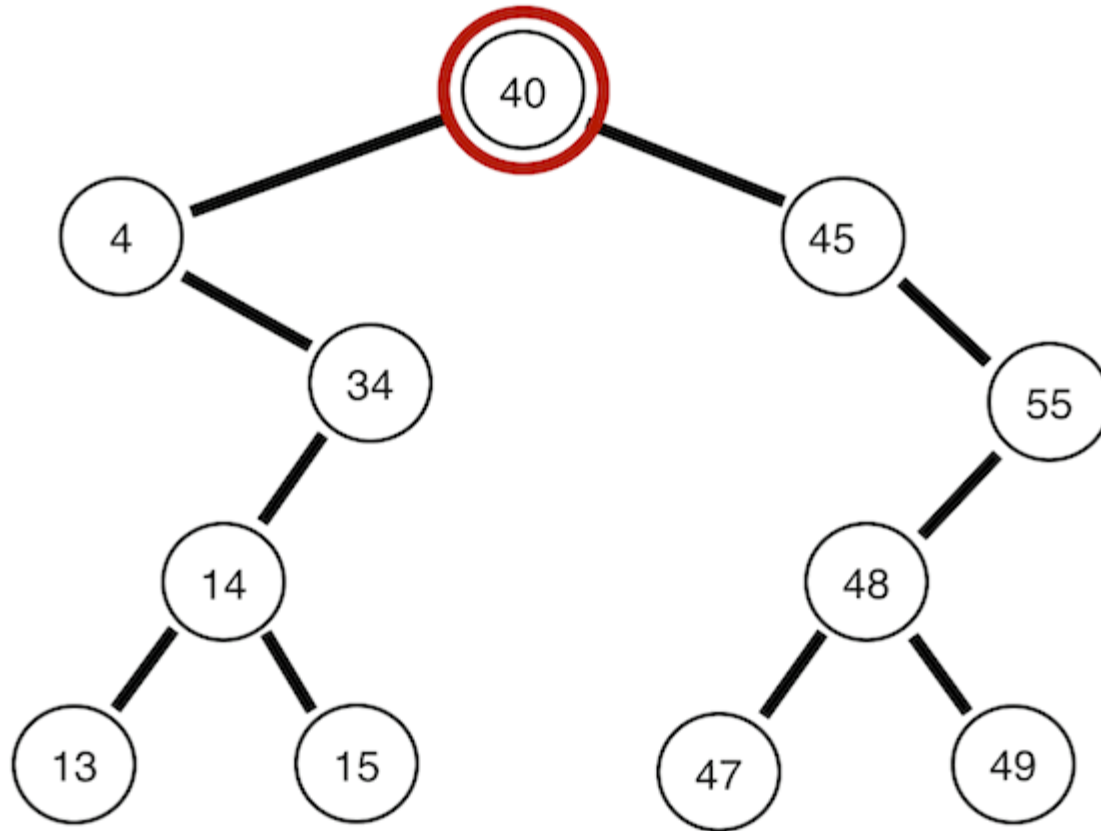
Traversals of a Binary Tree

- Use recursion
- To visit all the nodes in a binary tree, we must
 - Visit the root
 - Visit all the nodes in the root's left subtree
 - Visit all the nodes in the root's right subtree

Traversals of a Binary Tree

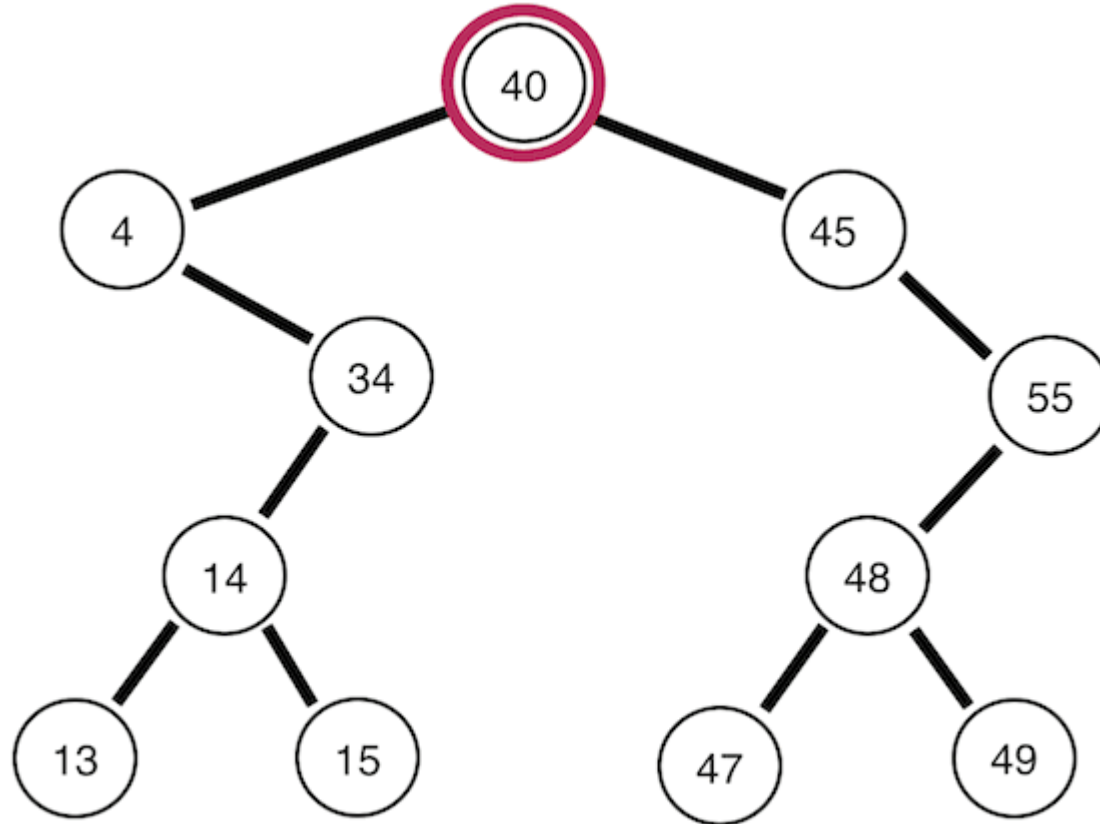
- Preorder traversal
 - Visit root, left subtree, right subtree
- Inorder traversal
 - Visit left subtree, root, right subtree
- Postorder traversal
 - left subtree, right subtree, root
- Level-order traversal
 - Begin at root and visit nodes one level at a time

Preorder Traversal of a Binary Tree



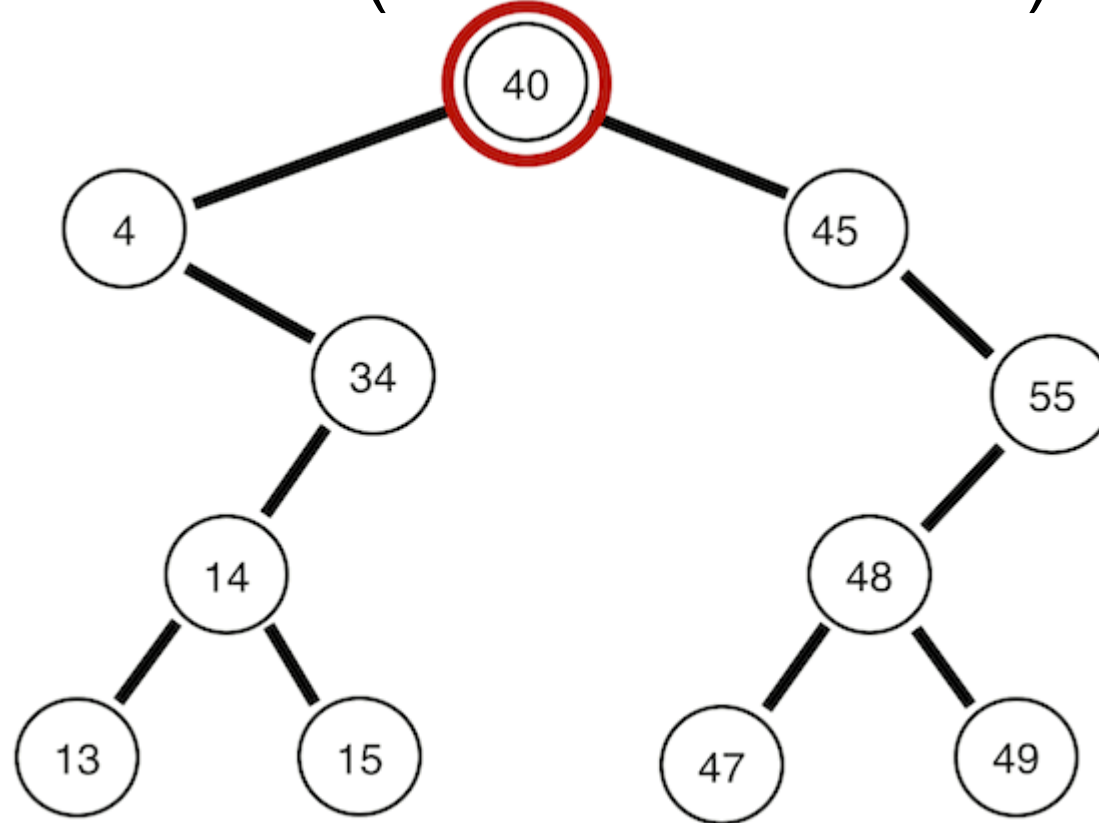
<https://austingwalters.com/binary-trees-traversals-everyday-algorithms/>

In-order Traversal of a Binary Tree



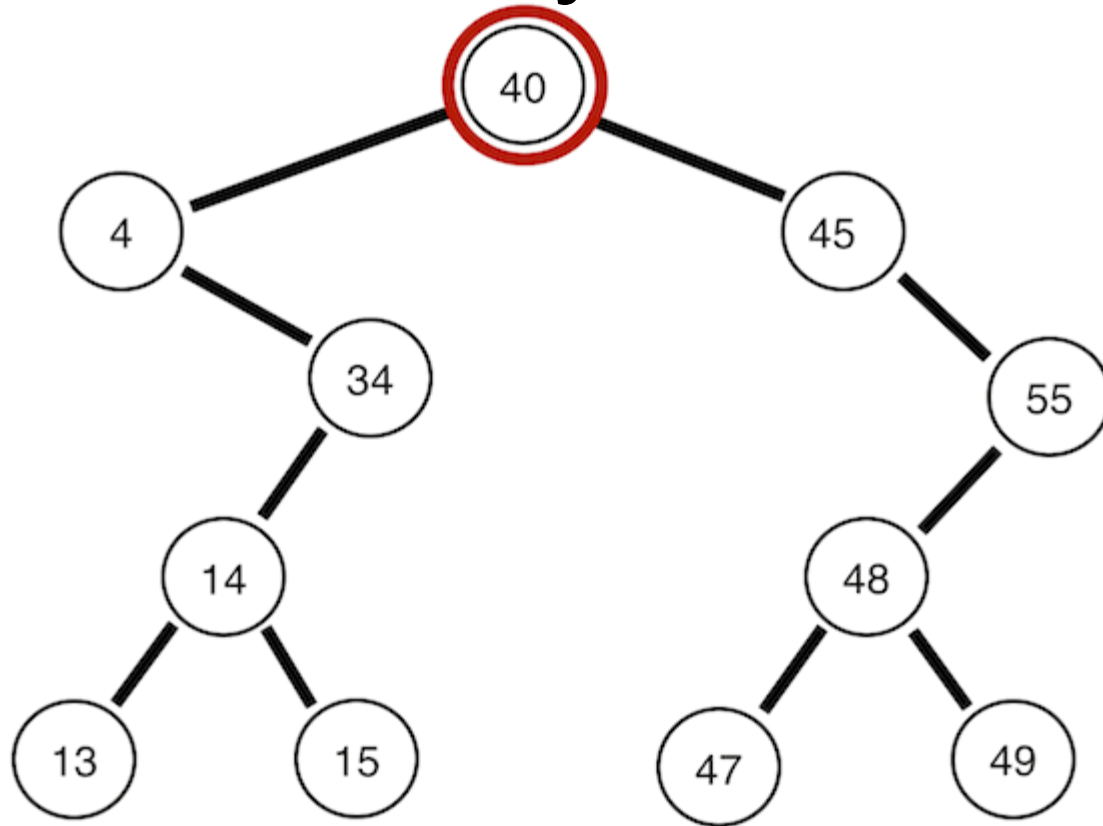
<https://austingwalters.com/binary-trees-traversals-everyday-algorithms/>

Post-order Traversal of a Binary Tree (variation from book)



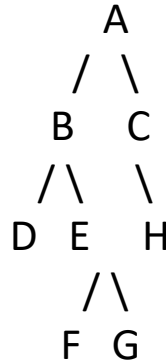
<https://austingwalters.com/binary-trees-traversals-everyday-algorithms/>

Level-order Traversal of a Binary Tree



<https://austingwalters.com/binary-trees-traversals-everyday-algorithms/>

Resulting Example Tree

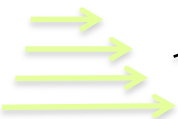


Root of tree is A
Height of tree is 4
nodes in tree is 8

Parent before

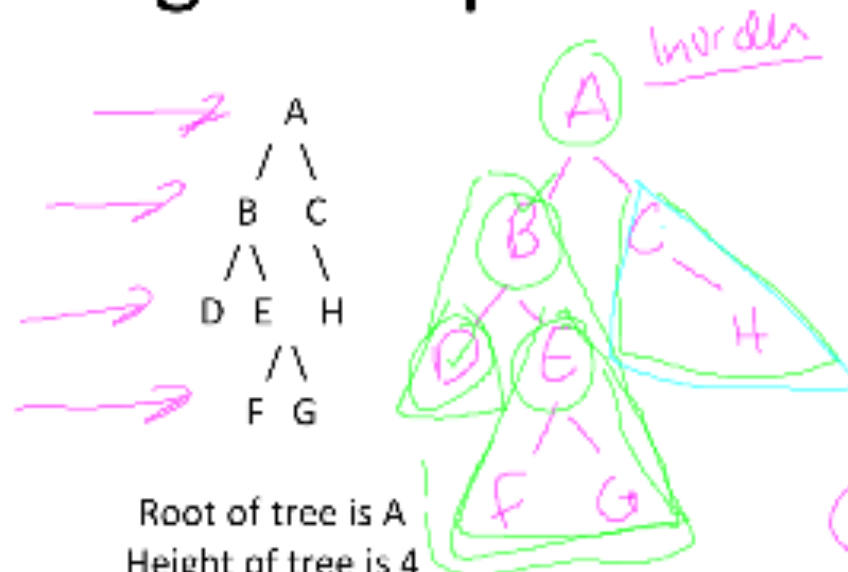


Parent after



Preorder:	A B D E F G C H
InOrder:	D B F E G A C H
PostOrder:	D F G E B H C A
LevelOrder:	A B C D E H F G

Resulting Example Tree



Root of tree is A
 Height of tree is 4
 # nodes in tree is 8

Parent before



Parent after



Preorder: ABDEFGCH
 InOrder: DBFEGACH
 PostOrder: DFGEBHCA
 LevelOrder: ABCDEFGHFG

