Binary Tree Traversals

- each element present is "visited" (or accessed) at least once

- Number of possible traversals: not a fixed number, depends on the structure

Idea!!!

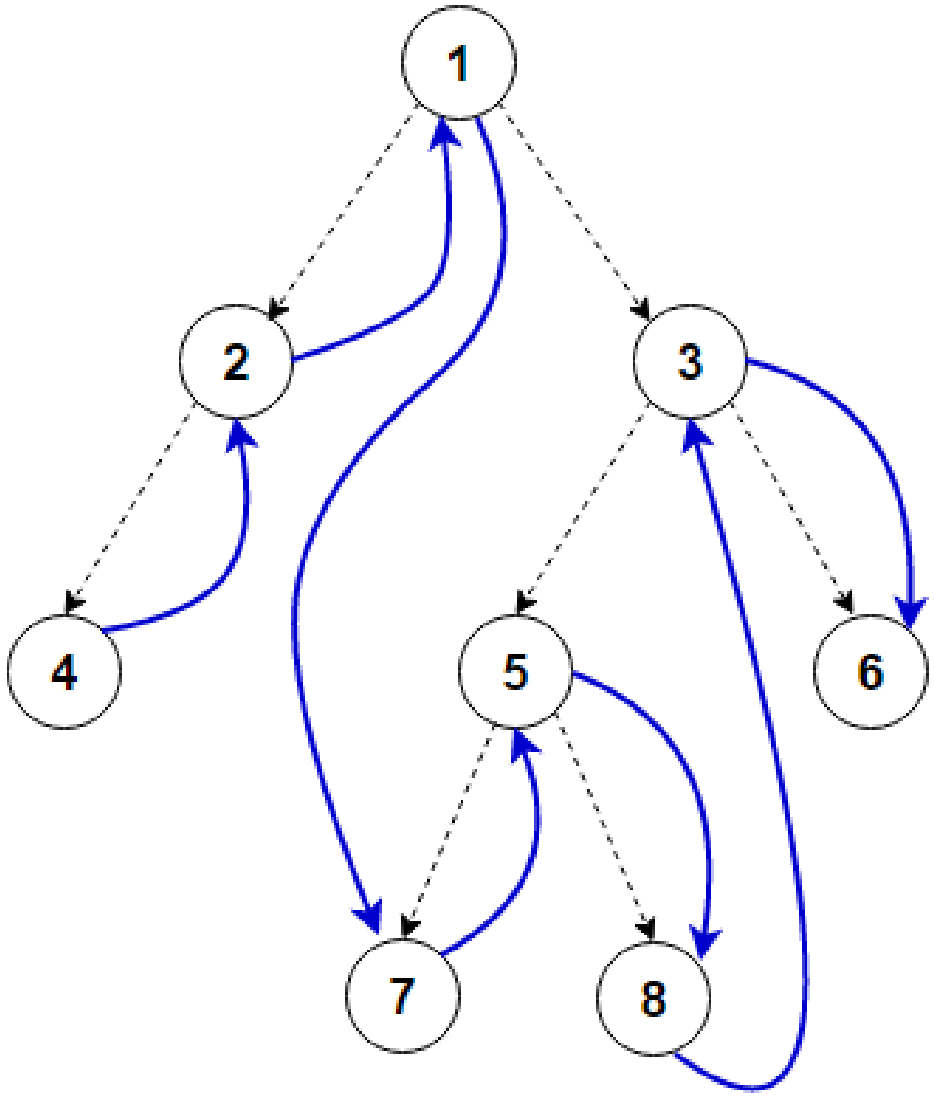
Recursion tree ☺

types of binary tree traversals

- Inorder
- Preorder
- Postorder

Inorder

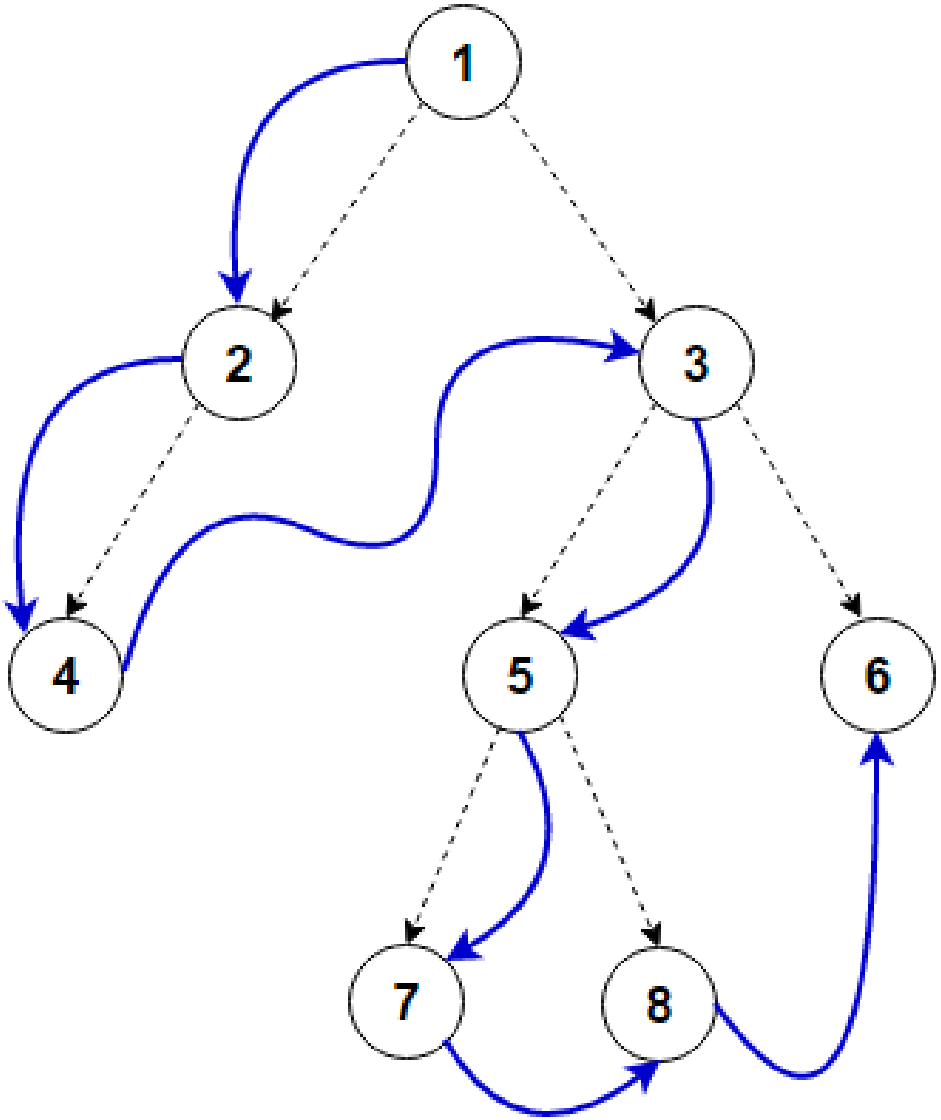
LVR (Left, Visit, Right)



Inorder: 4, 2, 1, 7, 5, 8, 3, 6

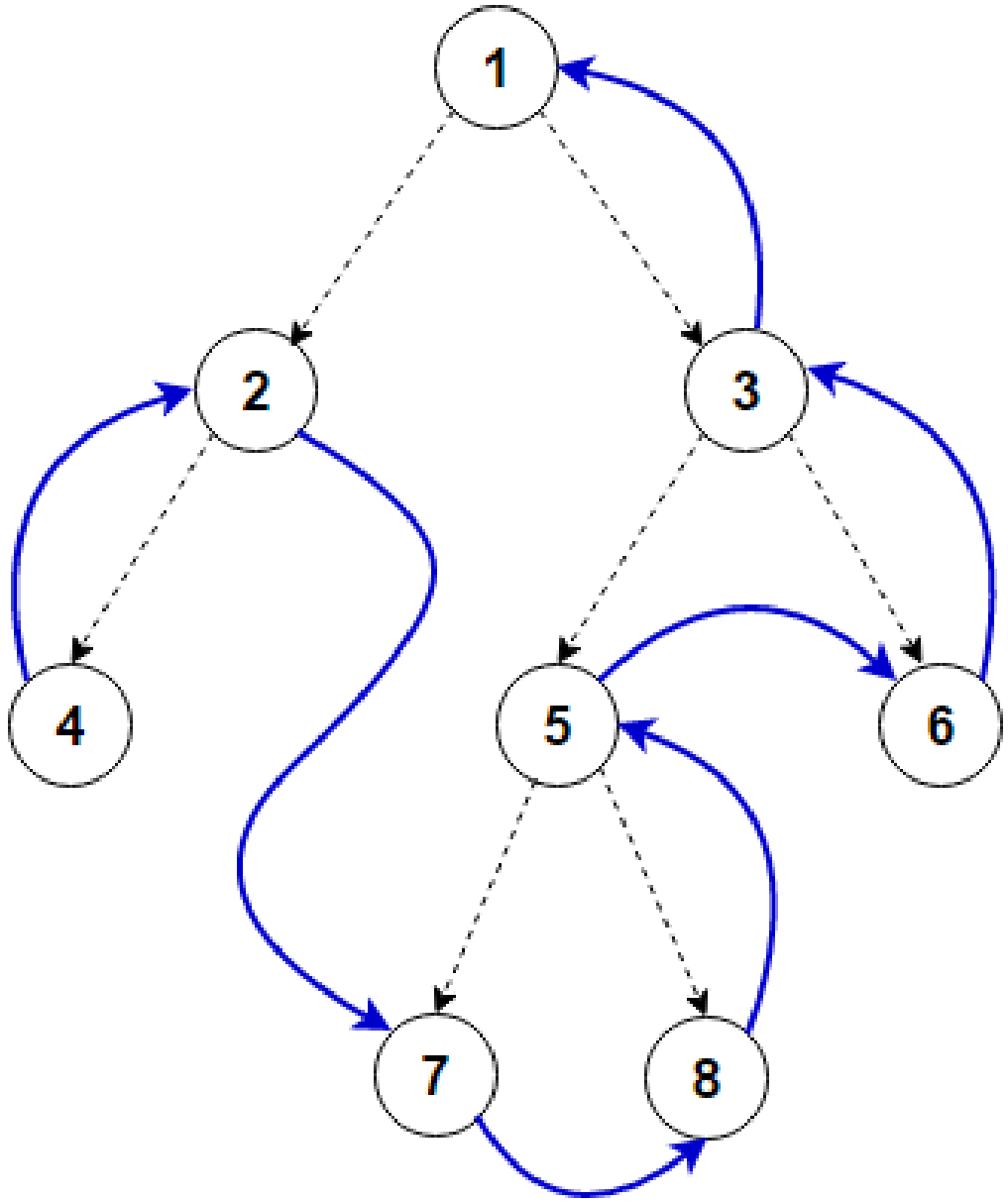
Preorder

VLR (Visit, Left, Right)



Postorder

LRv (Left, Right, Visit)



Example 1

Given the given Postorder and Inorder, construct its unique binary tree

PostOrder: DECBFA

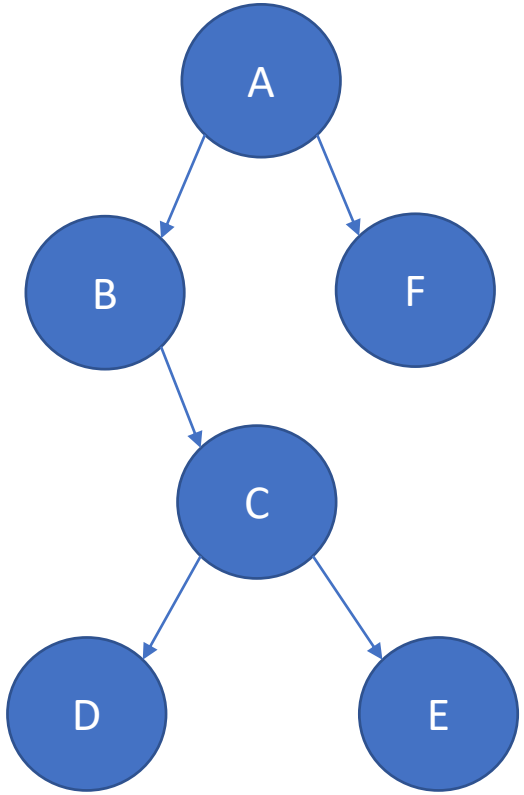
InOrder: BDCEAF

Example 1

Given the given Postorder and Inorder, construct its unique binary tree

PostOrder: DECBFA

InOrder: BDCEAF



Example 2

Given the given Postorder and Inorder, construct its unique binary tree

PostOrder: ABEFGHDC

InOrder: ABCEDGFH

Example 2

Given the given Postorder and Inorder, construct its unique binary tree

PostOrder: ABEFGHDC

InOrder: ABCEDGFH

Example 2

Given the given Postorder and Inorder, construct its unique binary tree

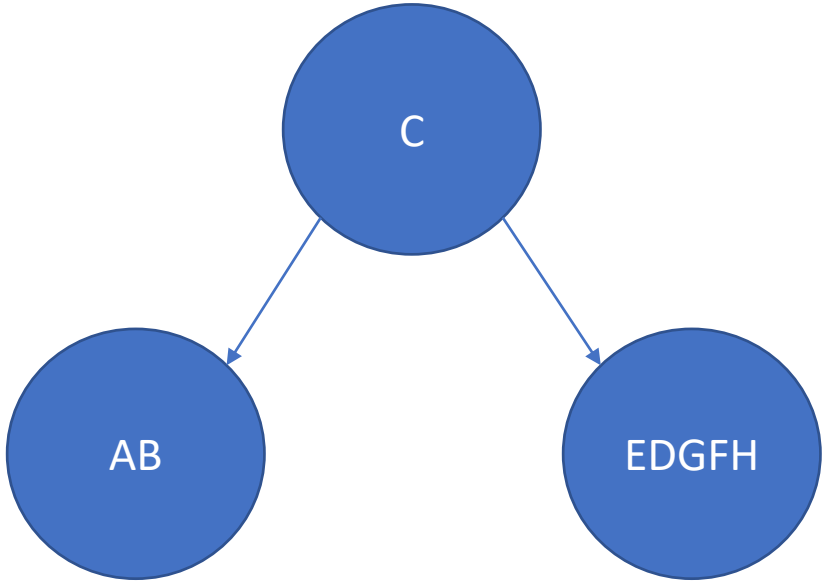
PostOrder: ABEFGHDC

Root

InOrder: AB C EDGFH

Visit

Left Right



Example 2

Given the given Postorder and Inorder, construct its unique binary tree

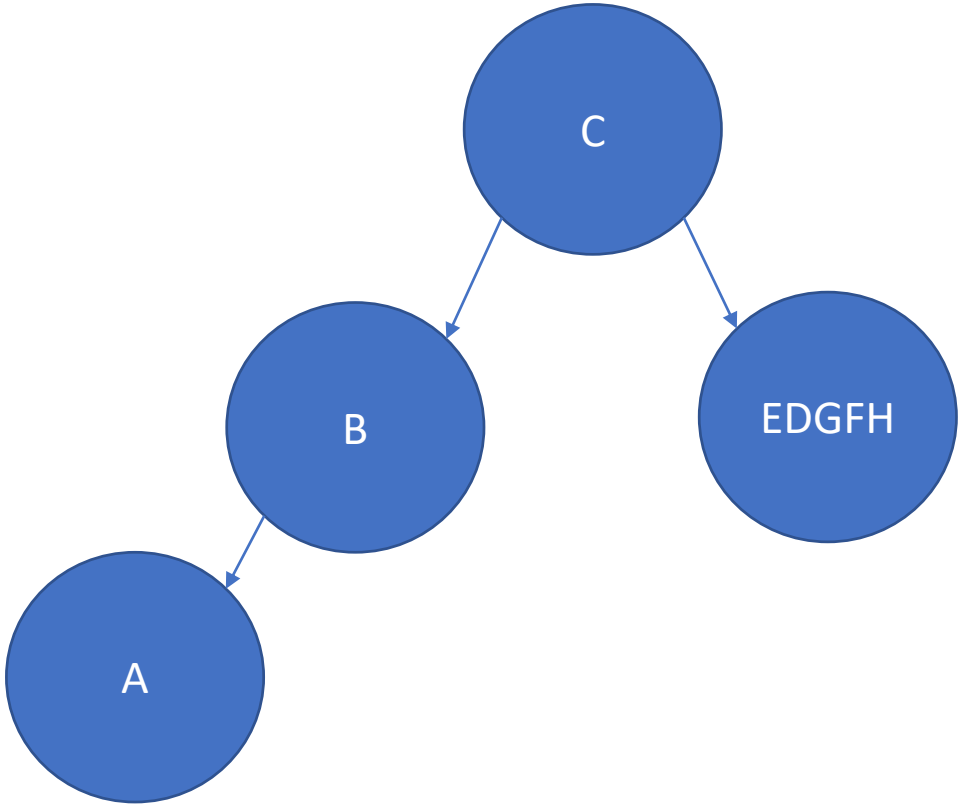
Root

PostOrder: AB

InOrder: A B ∅

Visit

Left Right



Example 2

Given the given Postorder and Inorder, construct its unique binary tree

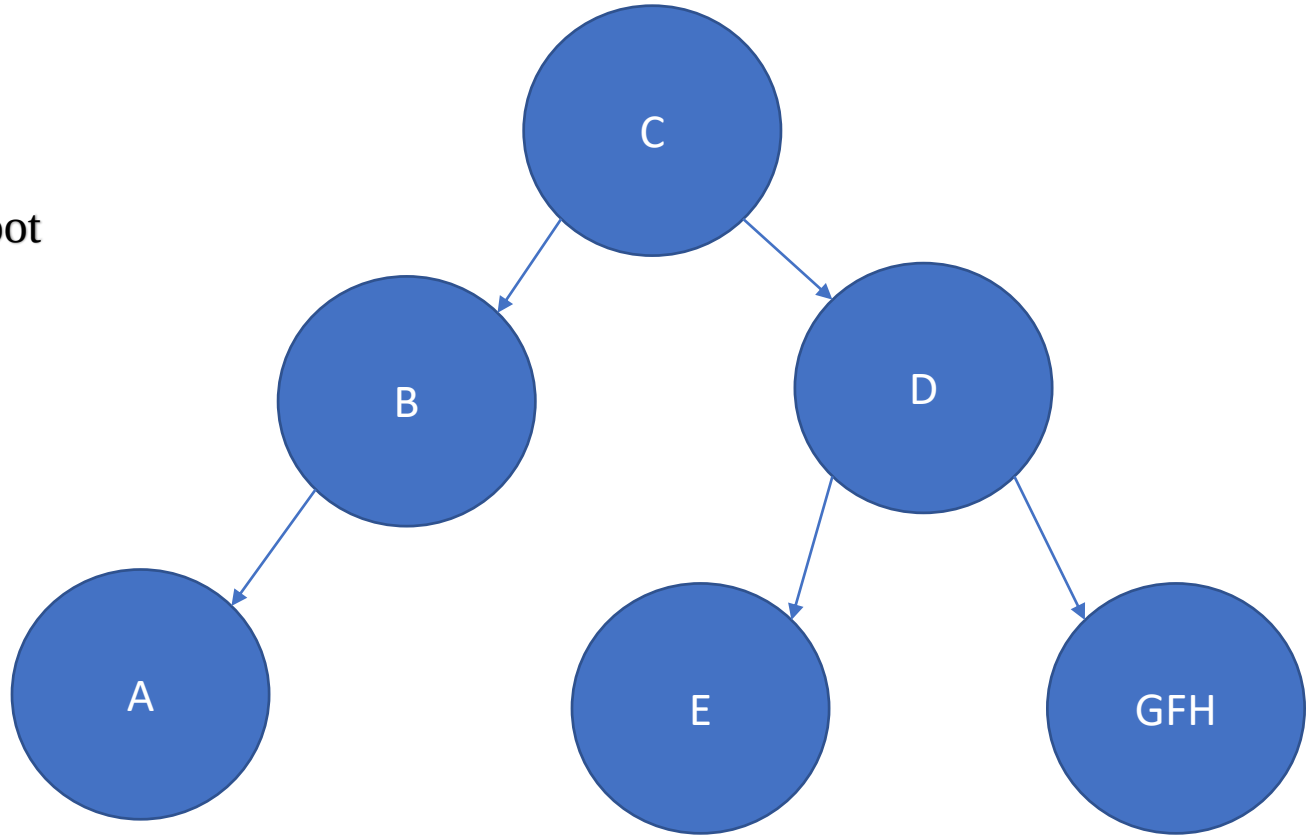
Root

PostOrder: EFGHD

InOrder: E D GFH

Visit

Left Right



Example 2

Given the given Postorder and Inorder, construct its unique binary tree

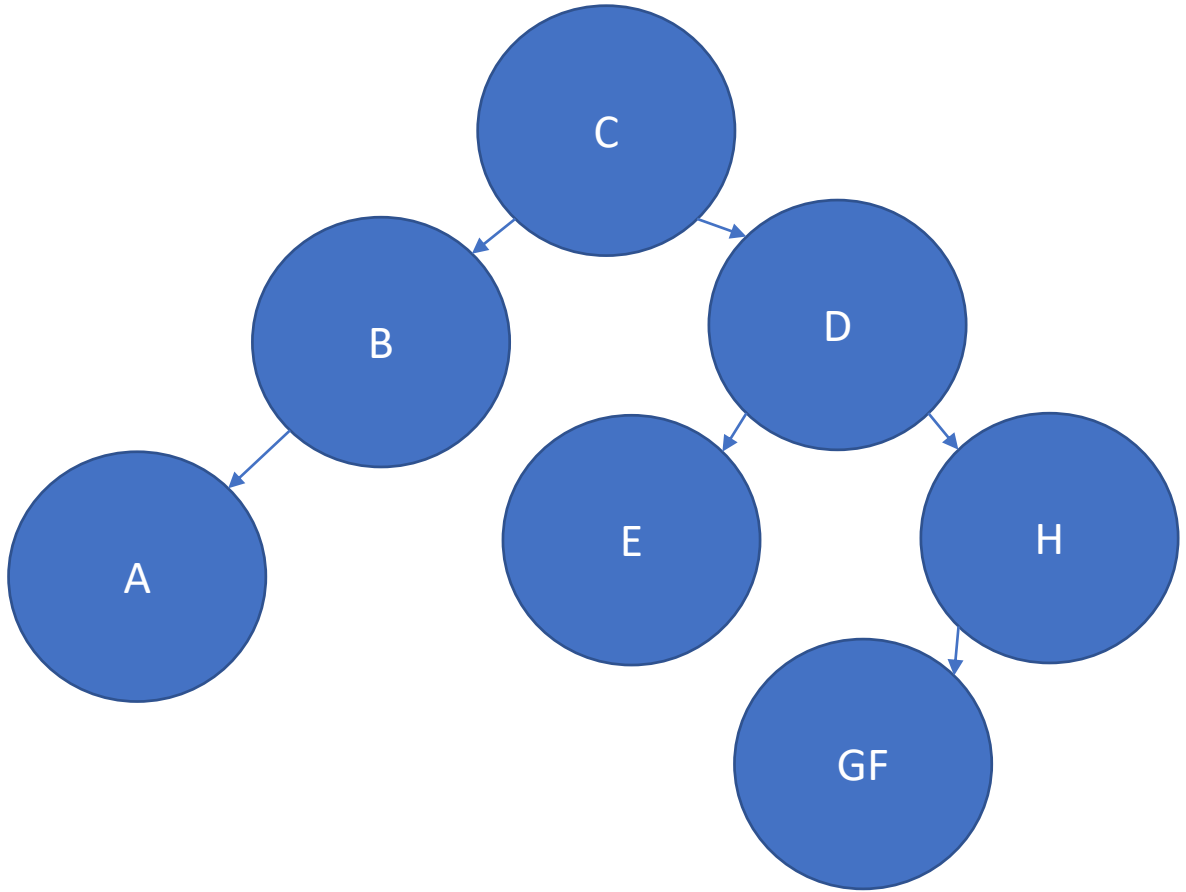
Root

PostOrder: FGH

InOrder: GF H Ø

Visit

Left Right



Example 2

Given the given Postorder and Inorder, construct its unique binary tree

Root

PostOrder: FG

InOrder: Ø G F

Visit

Left Right