Binary tree traversal

Ch 5.3, Ch 5.11

Some operations of binary trees

- Determine the height.
- Determine the number of nodes.
- Make a clone.
- Determine if two binary trees are clones.
- Display the binary tree.
- Evaluate the arithmetic expression represented by a binary tree.

HOW?

Binary Tree Traversal

- Visiting each node in the tree exactly once.
- When visiting a node, all action (copy, print, or count) with respect to this node is taken.
- A traversal produces a linear order for the nodes in a tree.
  - LVR, LRV, VLR, VRL, RVL, and RLV

L: moving left

V: visiting the node

R: moving right

Binary Tree Traversal methods

Depending on the position of the *visiting* (V):

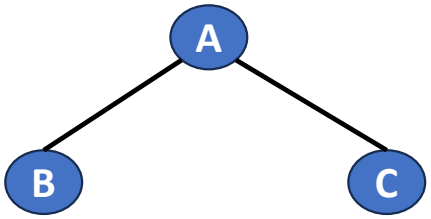
- LVR: inorder
- LRV: postorder
- VLR: preorder

Inorder traversal of a binary tree (LVR)

```
void inOrder(treePointer ptr)
{
    if (ptr != NULL)
    {
        inOrder(ptr->leftChild);
        visit(ptr);
        inOrder(ptr->rightChild);
    }
}
```

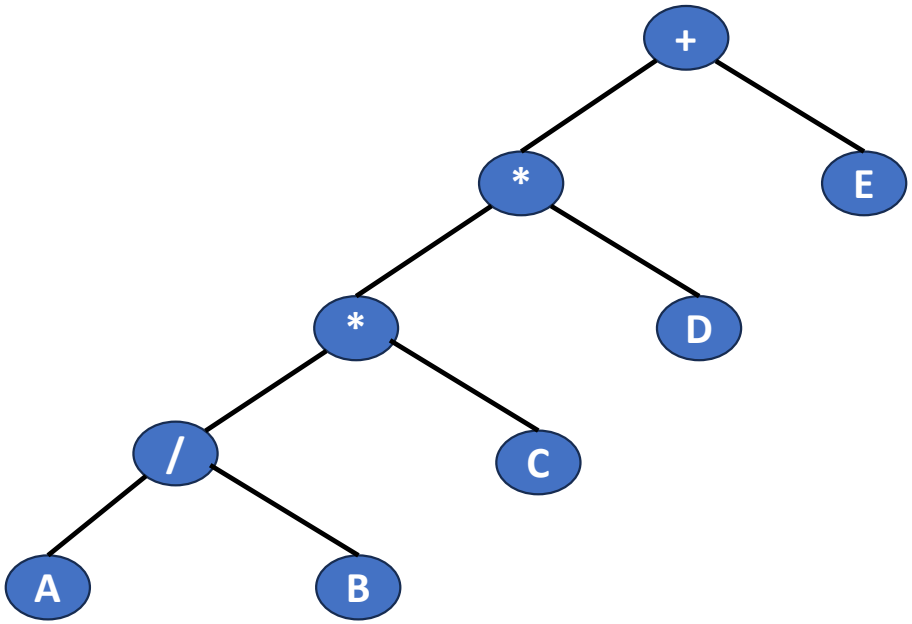
During visiting the node, the content will be printed.

Inorder example to print the tree



B A C

Binary tree with arithmetic expression



A / B * C * D + E

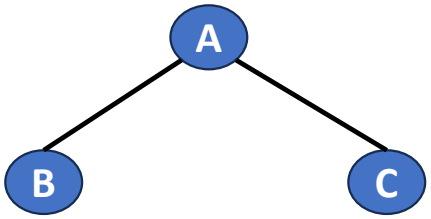
Infix form of expression

Preorder traversal of a binary tree (VLR)

```
void preOrder(treePointer ptr)
{
    if (ptr != NULL)
    {
        visit(t);
        preOrder(ptr->leftChild);
        preOrder(ptr->rightChild);
    }
}
```

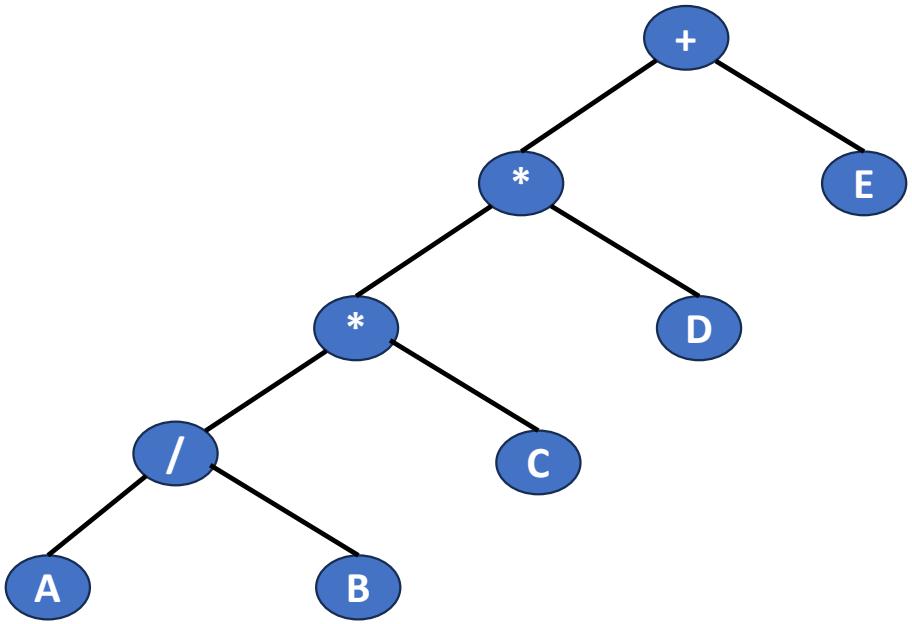
During visiting the node, the content will be printed.

Preorder example to print the tree



A B C

Binary tree with arithmetic expression



+ * * / A B C D E

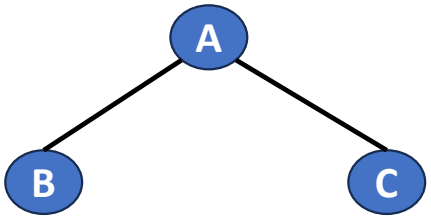
Prefix form of expression

Postorder traversal of a binary tree (LRV)

```
void postOrder(treePointer ptr)
{
    if (ptr != NULL)
    {
        postOrder(ptr->leftChild);
        postOrder(ptr->rightChild);
        visit(t);
    }
}
```

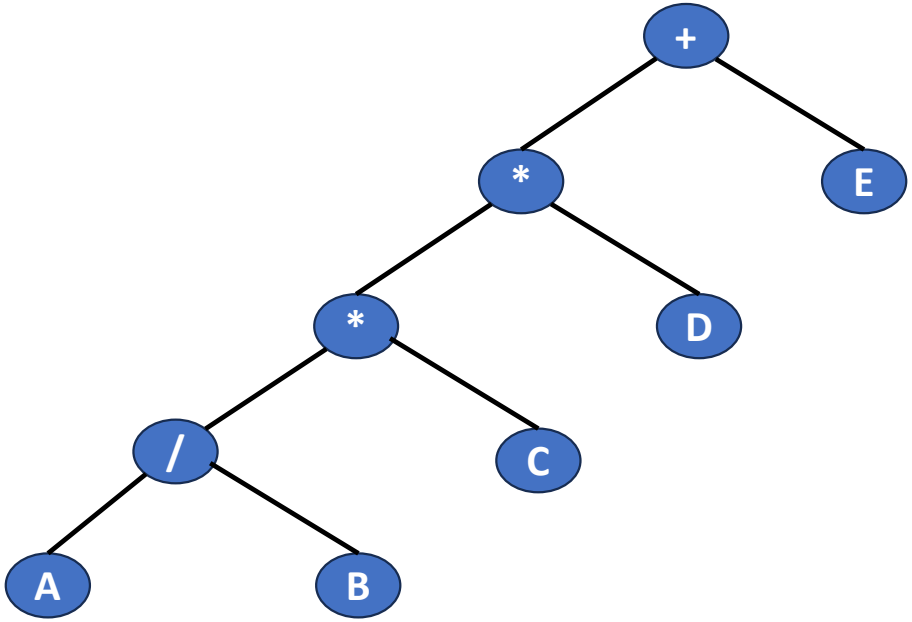
During visiting the node, the content will be printed.

Postorder example to print the tree



B C A

Binary tree with arithmetic expression



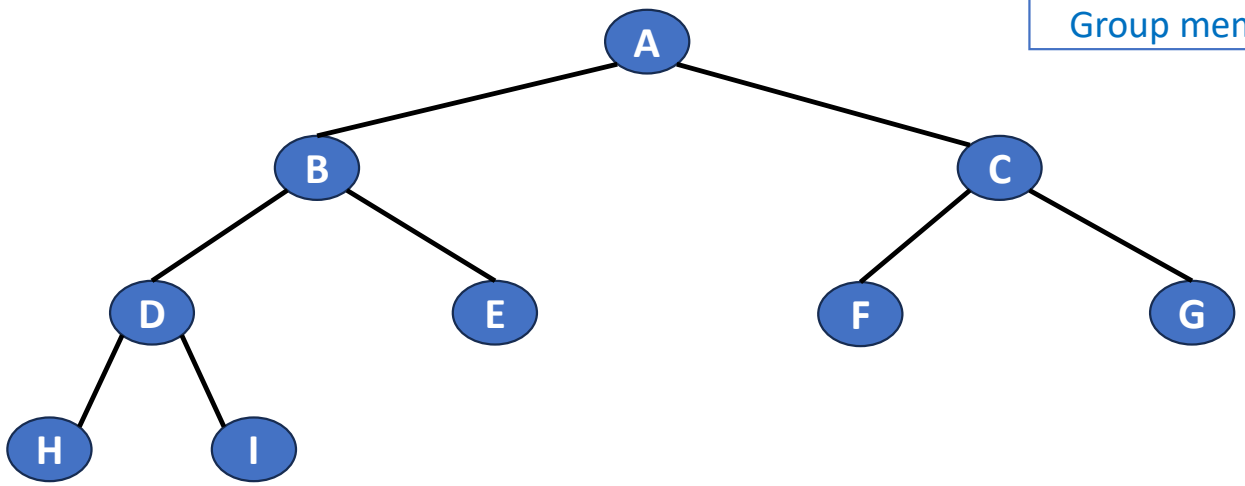
A B / C * D * E + **Postfix form of expression**

Exercise

- Q1: Write out the inorder traversal
- Q2: Write out the preorder traversal
- Q3: Write out the postorder traversal

Please reply your answers of Q1-Q3 via the following link:

Group members: 1~3 people

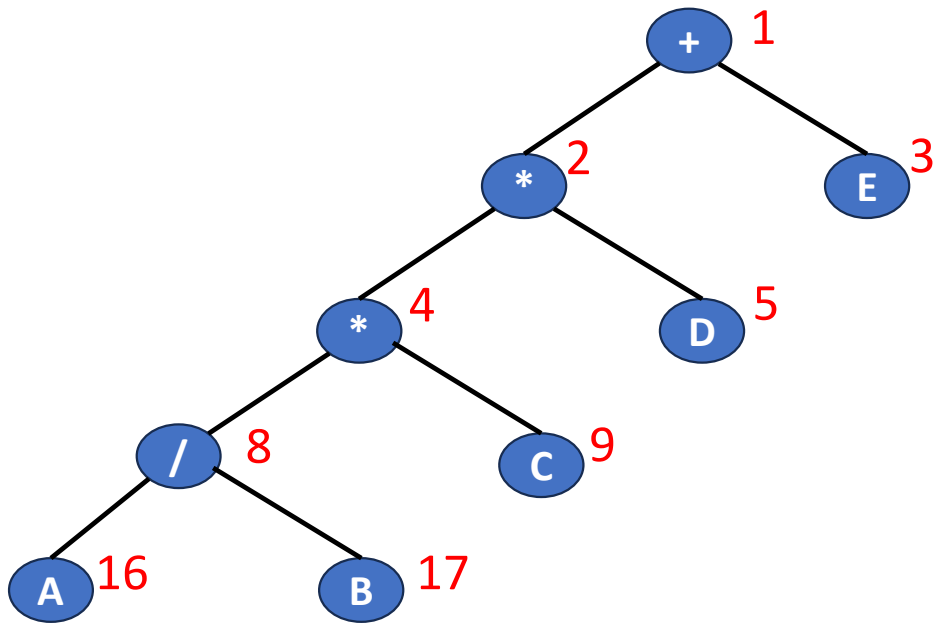


Level-order traversals

- Visiting the nodes following the order of node numbering scheme (sequential numbering)
  - Visit the root first
  - Visit the root’s left child
  - Visit the root’s right child
  - Go to the next new level
  - Visit from the leftmost node to the right most node

Level-order example to print the tree

Binary tree with arithmetic expression



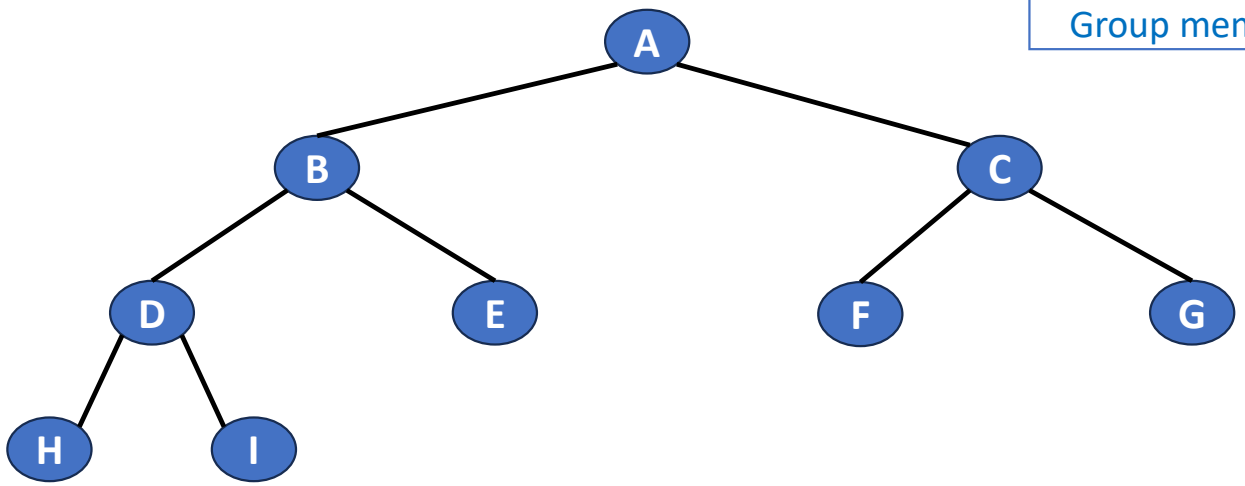
+ * E * / C A B

Exercise

- Q4: Write out the level-order traversal.

Please reply your answers of Q4 via the following link:

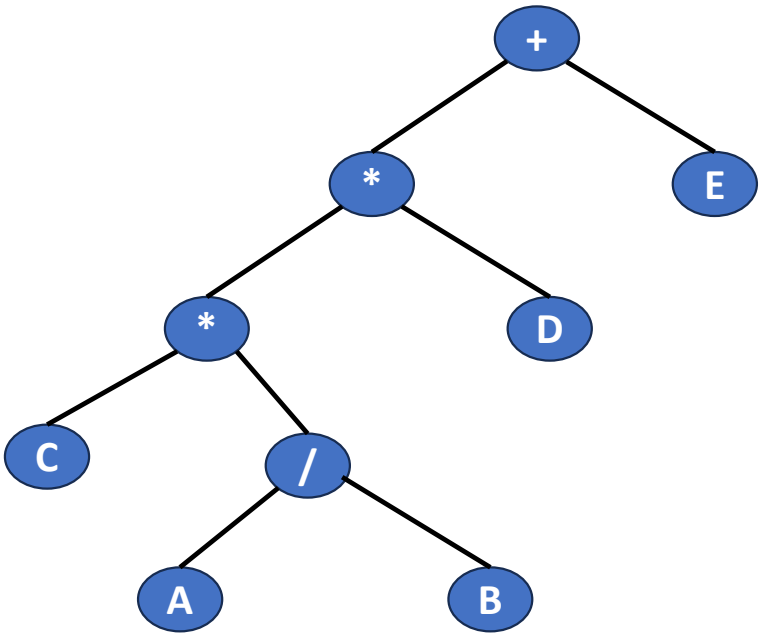
Group members: 1~3 people

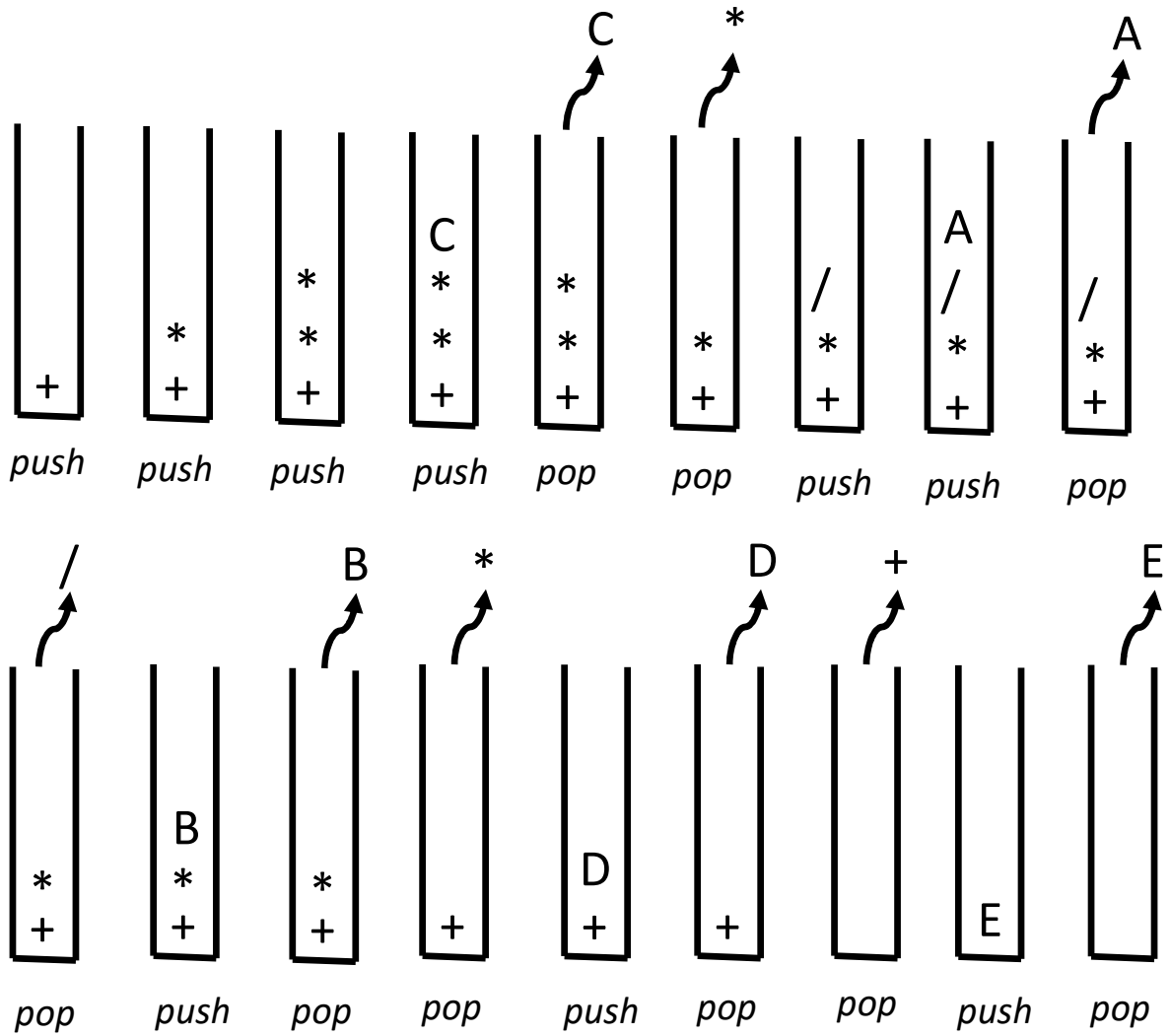


Iterative inorder traversal

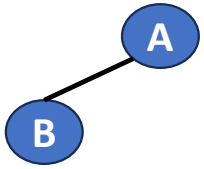
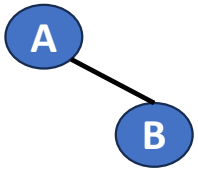
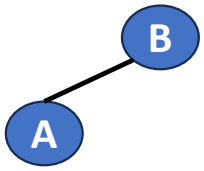
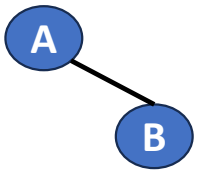
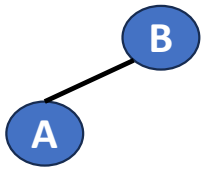
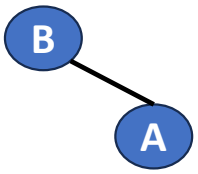
- Left nodes are stacked until a null node is reached.
- Pop the top node from the stack.
- The right child of the pop-out node is stacked.
- Continue the traversal from the right child.

Example


+
*
E
*
D
C
/
A
B


push
push
push
push
pop
pop
push
push
pop
pop
push
pop
pop
push
pop
pop
push
pop
C
*
A
/
B
*
D
+
E

One sequence gives distinct binary trees.

Given sequence:	**Potential trees:**	
preorder AB		
inorder AB		
postorder AB		

Uniquely defining a binary tree

Given sequences:	**Potential trees:**
preorder AB *postorder* BA	 **Not unique**
preorder AB *inorder* BA	 **Unique**

Constructing a unique binary tree from inorder and preorder sequences

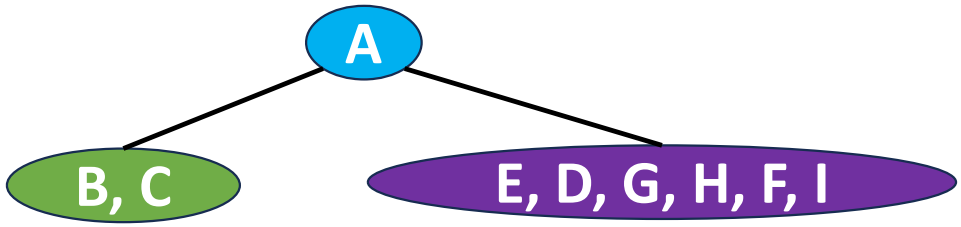
root

Preorder = A B C D E F G H I

Inorder = B C A E D G H F I

left *right*

- Scan the preorder from left to right to find a root
- Scan the inorder to separate left and right subtrees



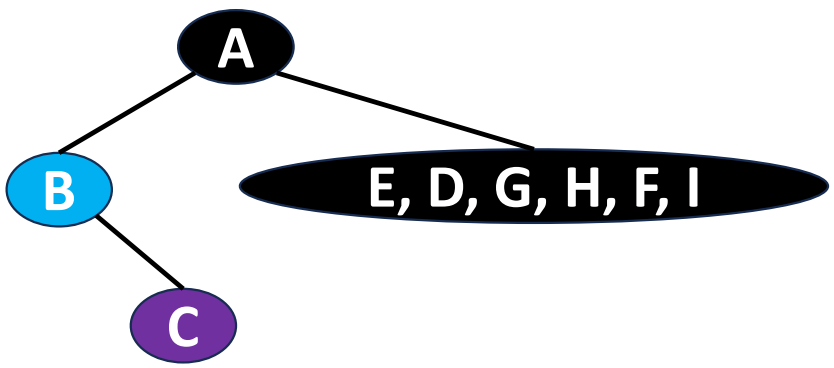
Constructing a unique binary tree from inorder and preorder sequences

root

Preorder = A B C D E F G H I

Inorder = B C A E D G H F I

right

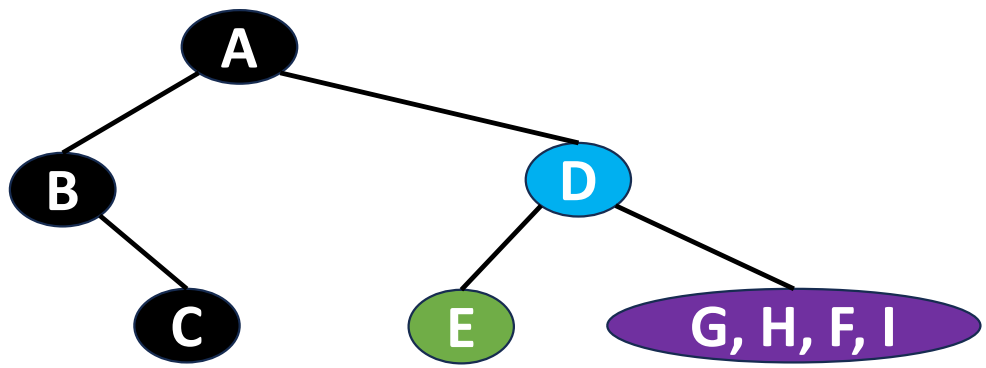


root

Preorder = A B C D E F G H I

Inorder = B C A E D G H F I

left *right*



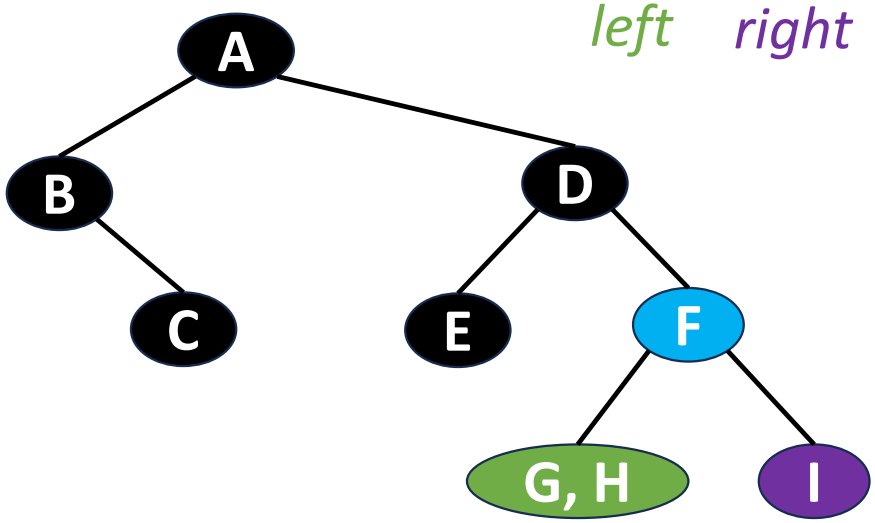
Constructing a unique binary tree from inorder and preorder sequences

root

Preorder = A B C D E F G H I

Inorder = B C A E D G H F I

left *right*

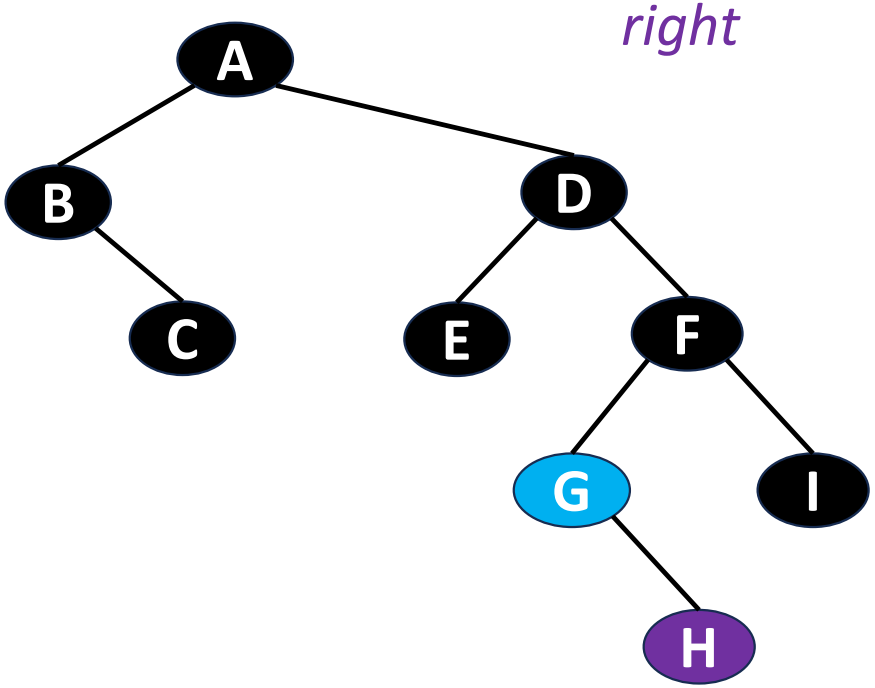


root

Preorder = A B C D E F G H I

Inorder = B C A E D G H F I

right



Exercise

- Plot the binary tree from the following two sequences:

Postorder = G H D I E B J F C A
Inorder = G D H B E I A F J C

Hint: Direction to scan postorder is from right to left.

Summary

- Traversal methods:
  - Inorder, preorder, postorder, level-order
- Recursive and iterative traversal
- Constructing a binary tree using two types of sequences