

# Data Structure Lecture 8: Tree 2

Prepared by  
Dr. Mohammed Salah Al-Obiadi

# Traversing With Tree

- The tree traversing is the way to visit all the nodes of the tree on a specific order.
- The Tree traversal can be accomplished in four different ways:
  - Inorder Traversal
  - Pre Order Traversal.
  - Post Order Traversal
  - Level Order Traversal

# Inorder traversal

- Traverse the Left Subtree in **INORDER(Left)**
- Visit the Root node
- Traverse the Right Subtree in **INORDER(Right)**

# Preorder Traversal

- Visit the Root Node
- Traverse the Left Subtree in **PREORDER(Left)**
- Traverse the Right Subtree in **PREORDER(Right)**

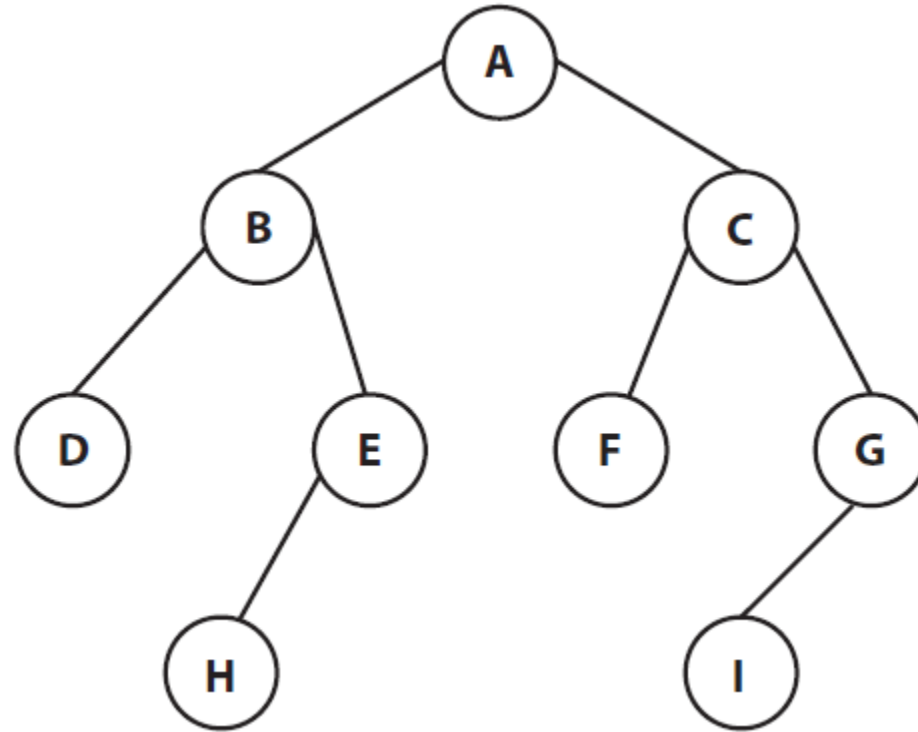
# Postorder Traversal

- Traverse the Left Subtree in **POSTORDER**(Left)
- Traverse the Right Subtree in **POSTORDER**(Right)
- Visit the Root Node

# Level Order Traversal

- In this type of traversal the elements will be visited according to level wise but it is not so far used.

# Examples



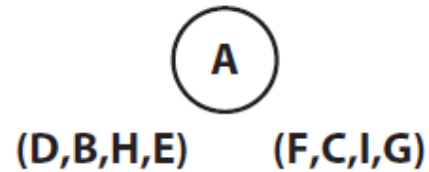
1. Inorder : **D B H E A F C I G**
2. Preorder : **A B D E H C F G I**
3. Post Order : **D H E B F I G C A**
4. Level Order : **A B C D E F G H I**

# Conversion Of A Tree From Inorder And Preorder

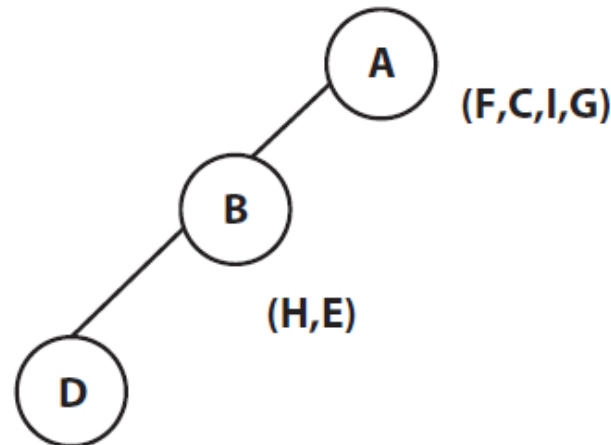
- **INORDER : D B H E A F C I G**

- **PREORDER : A B D E H C F G I**

- Choose the **ROOT** from the preorder and from inorder find the nodes in left and right and this process will continue up to all the elements are chosen from the preorder/inorder.
- **STEP1:** From preorder A is the root and from inorder we will find that in the left of A (**D,B,H,E**) and in the right (**F,C,I,G**):

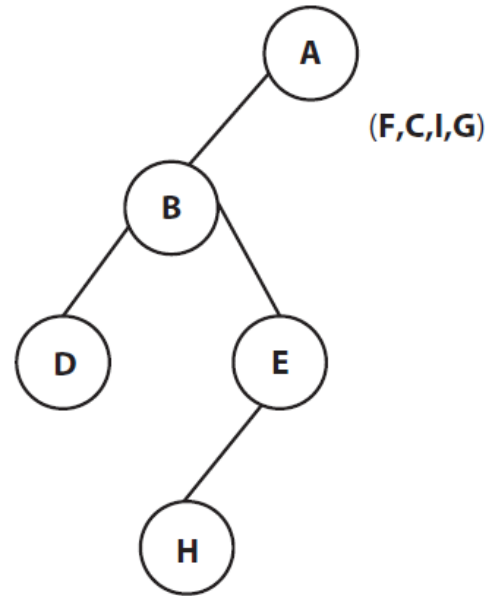


- **STEP2:** Again from Preorder 'B' will be chosen as **PARENT** and from Inorder in the left of B (**D**) and in the right (**H,E**):

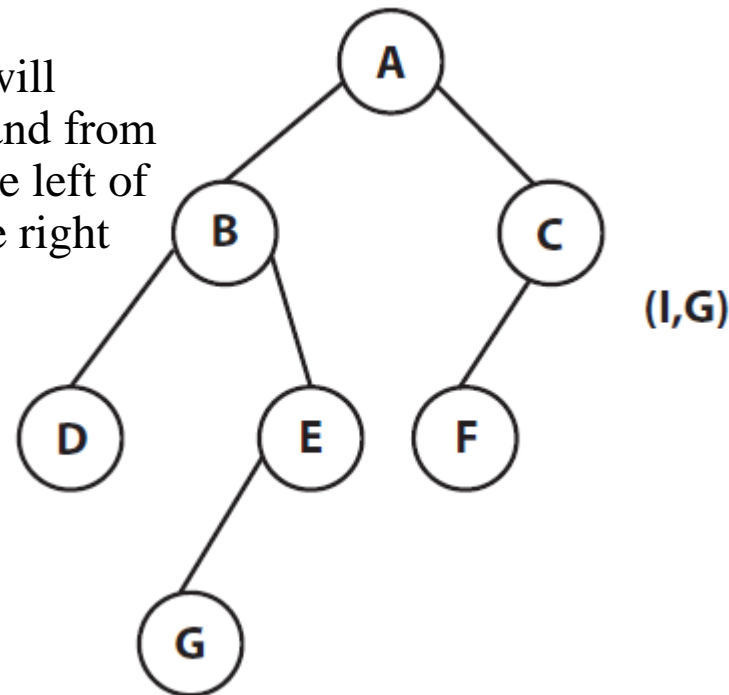




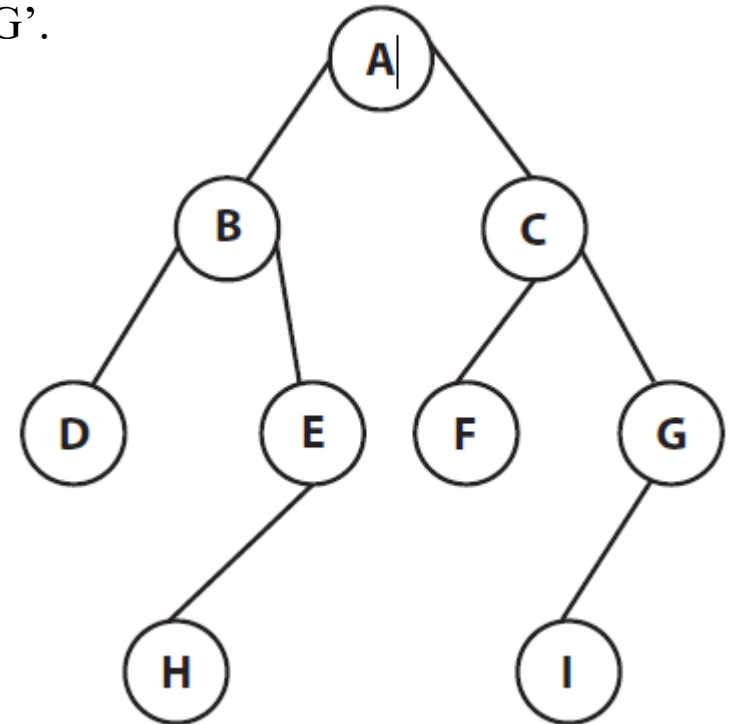
- **STEP3:** From Preorder 'E' will chosen as the PARENT and from inorder on its left 'H' is present:



- **STEP4:** From Preorder we will choose 'C' as the PARENT and from inorder we observe that in the left of 'C' (F) will placed and in the right (I,G)

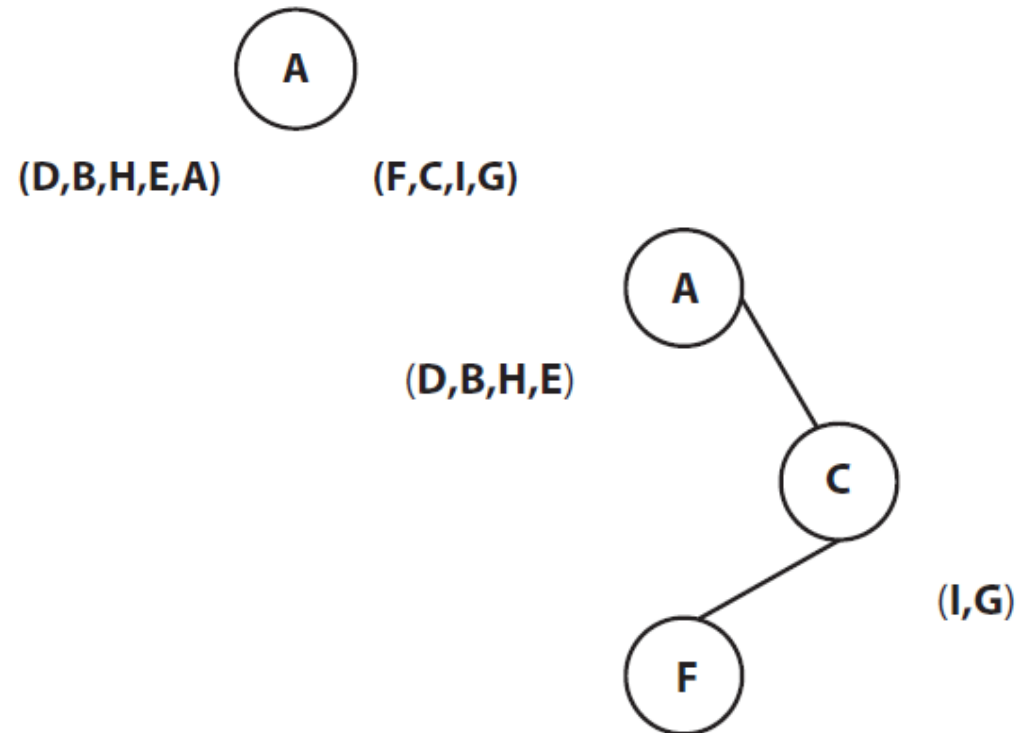


- **STEP5:** From the PREORDR we observe that 'G' is the parent and from the INORDER I will be used as the Left child of 'G'.



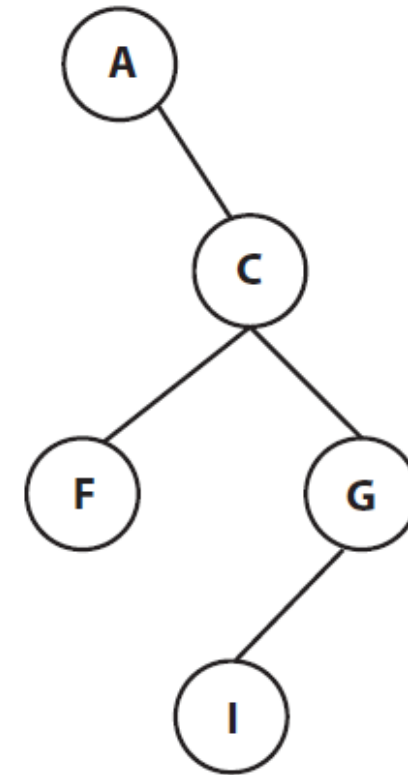
## Conversion Of A Tree From Inorder And Postorder

- **INORDER : D B H E A F C I G**
- **POST ORDER : D H E B F I G C A**
- Choose the **ROOT** from the postorder (**from the right**) and from inorder find the nodes in left and right and this process will continue up to all the elements are chosen from the postorder/inorder.
  - **STEP1 :** From the right of POSTORDER 'A' will be chosen as the ROOT and from INORDER we observe that in the left of A (**D,B,H,E,A**) and in the right (**F,C,I,G**) will be there.
  - **STEP2:** From the POSTORDER 'C' will be chosen as the PARENT and from INORDER we observe that in the right of 'C' (**I,G**) and to the left (**F**) will be used.

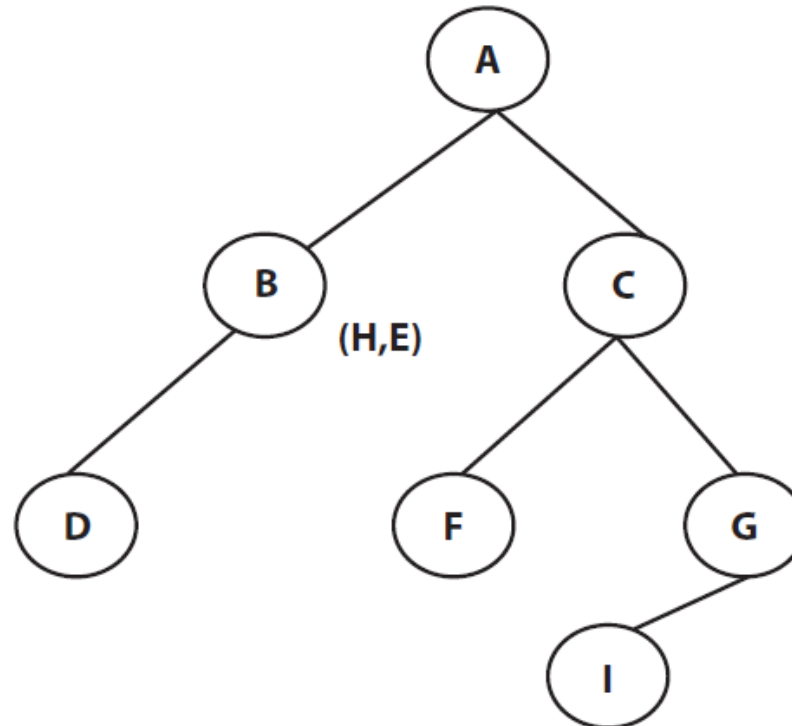


- **STEP3:** From the right of POSTORDER 'G' will be chosen as the PARENT and from inorder to the left of 'G' (I) will be used.

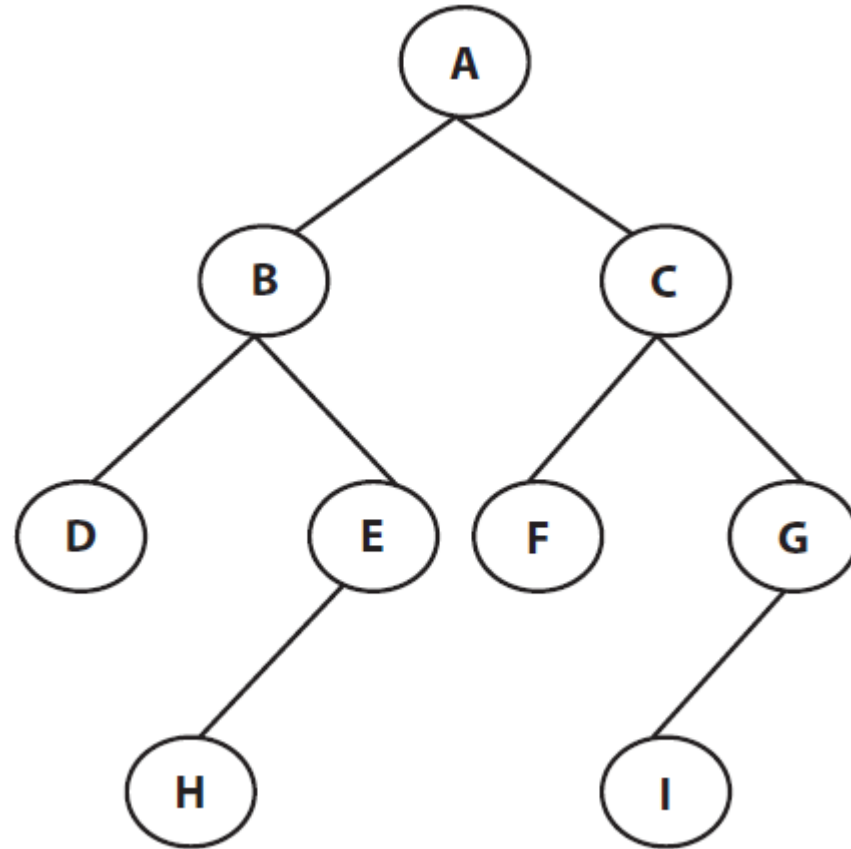
(D,B,H,E)



- **STEP4:** From the right to postorder 'B' will be chosen as the PARENT and from the INORDER to we observe that to the right of 'B' (H,E,A) and to the left (D) will be used.



- **STEP5:** From the right to postorder we will choose 'E' as PARENT and from the Inorder to the left of 'E' (**H**) will be used.



# Applications of Binary Tree

- Expression Tree
- Binary Search Tree
- Height Balanced Tree (AVL Tree)
- Threaded Binary Tree
- Heap Tree
- Huffman Tree
- Decision Tree
- Red Black Tree

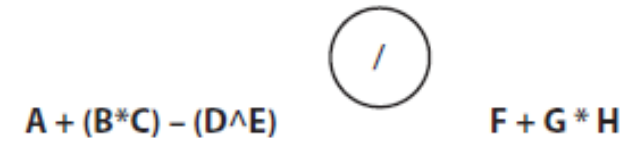
# Expression Tree

- An expression tree is a Binary Tree which stores/represents the mathematical (arithmetic) expressions.
- The leaves of an expression tree are operands, such as constants or variable names and all the internal nodes are the operators.
- Formally we can define an expression Tree as a special kind of binary tree in which:
  - Each leaf is an operand. Examples: a, b, c, 6, 100
  - The root and internal nodes are operators. Examples: +, -, \*, /, ^
  - Subtrees are subexpressions with the root being an operator.

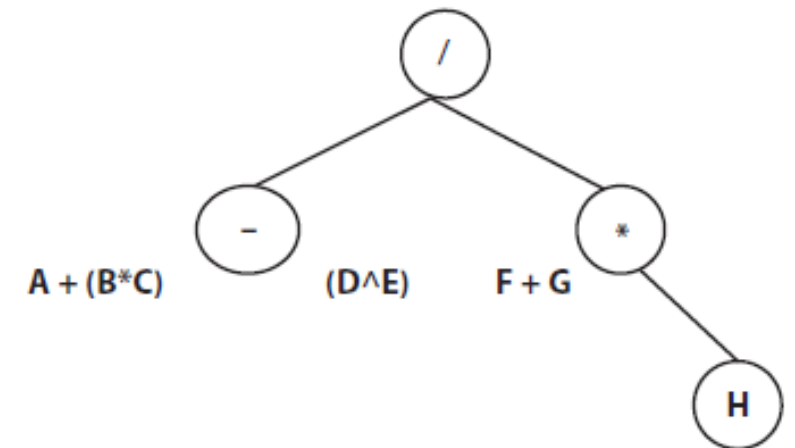
# EXAMPLE

- Represent an Expression Tree
  - $A + (B * C) - (D \wedge E) / F + G * H$
- choose an operator in such a way that the terms in parenthesis will be in a side
- Choose a operator having higher precedence.

STEP1:



STEP2:



Step3:

