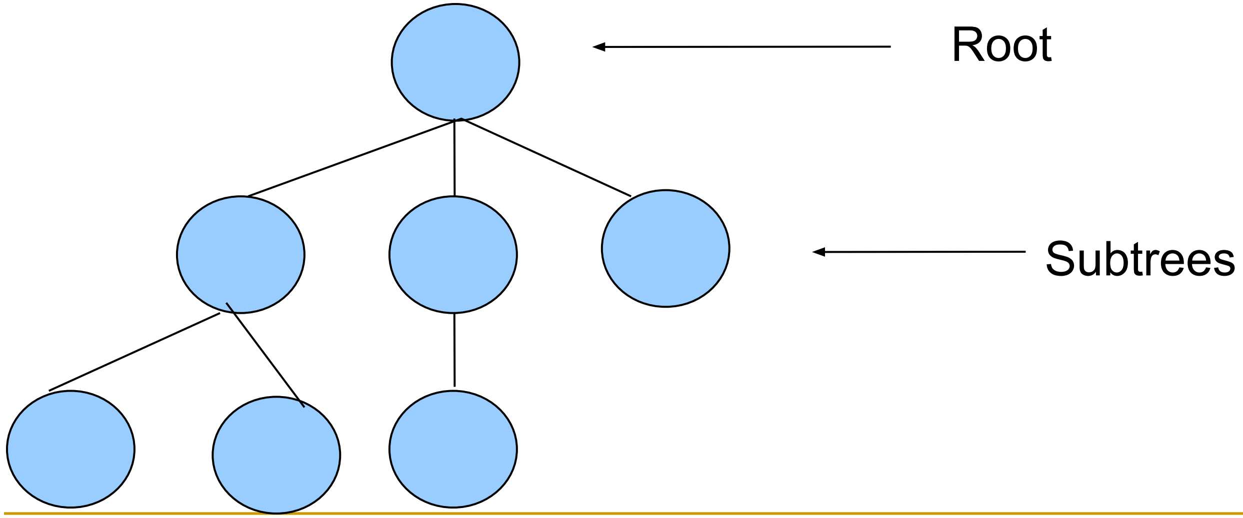
Data Structure
CSE-207

Tree

Tree

- A nonlinear data structure
- Contain a distinguished node R, called the root of tree and a set of subtrees.
- Two nodes n1 and n2 are called siblings if they have the same parent node.

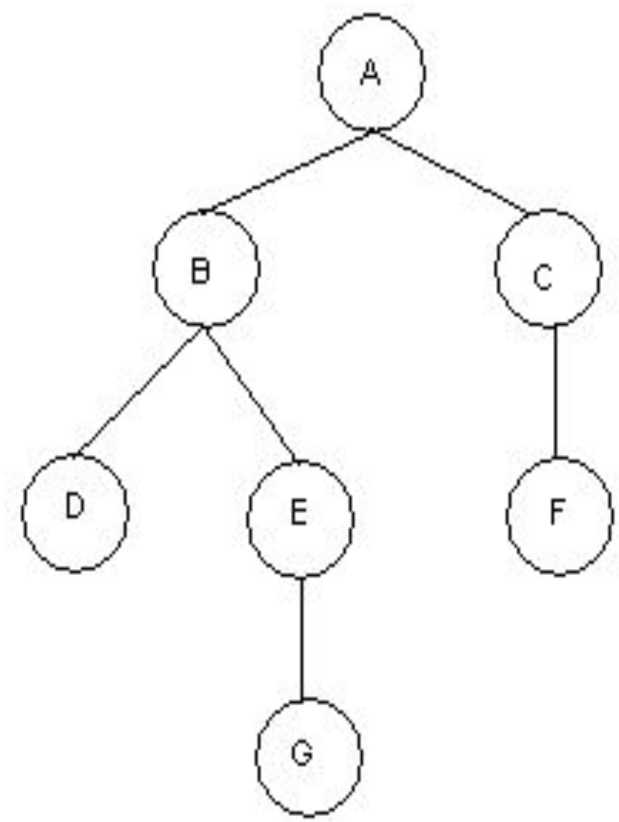


Binary Tree

- A binary tree T is defined as a finite set of elements, called nodes such that:
  - T is empty
  - T contains a distinguished node R, called the root of T and the remaining nodes of T form an ordered pair of disjoint binary trees T1 and T2.
  - T1 and T2 are called the left and right subtrees of R.

- Any node N in a binary tree T has either 0, 1 or 2 successors.
- Nodes with no successors are called terminal nodes or leaf nodes.

Binary Tree



- Binary Tree: T
- Root: A
- Nodes with 2 Successors: A, B
- Nodes with 1 Successors: C, E
- Terminal Nodes: D, F, G

Binary Tree

- Similar binary tree
  - Two binary trees are similar if they have the same structure or same shape.
- Copy Binary Trees
  - Two binary trees are copies if they are similar and they have the same contents at the corresponding nodes.

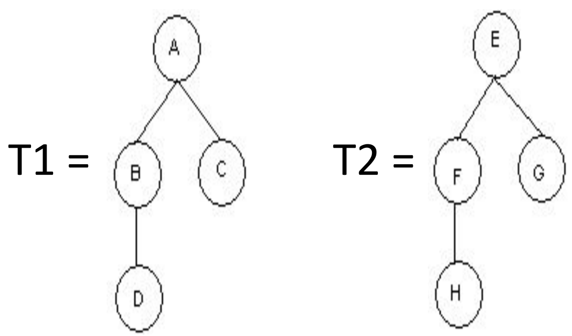


Figure: Similar T1 and T2.

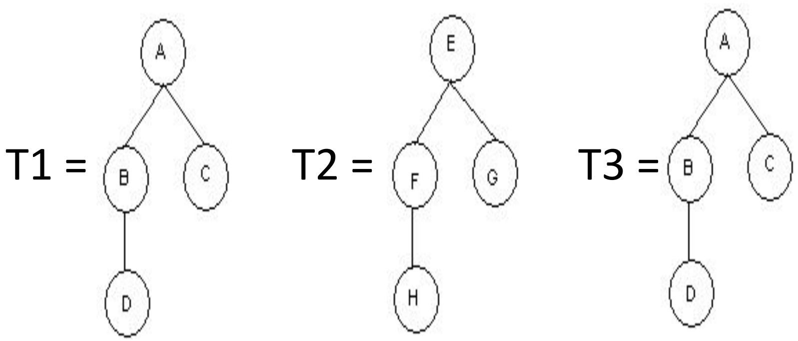
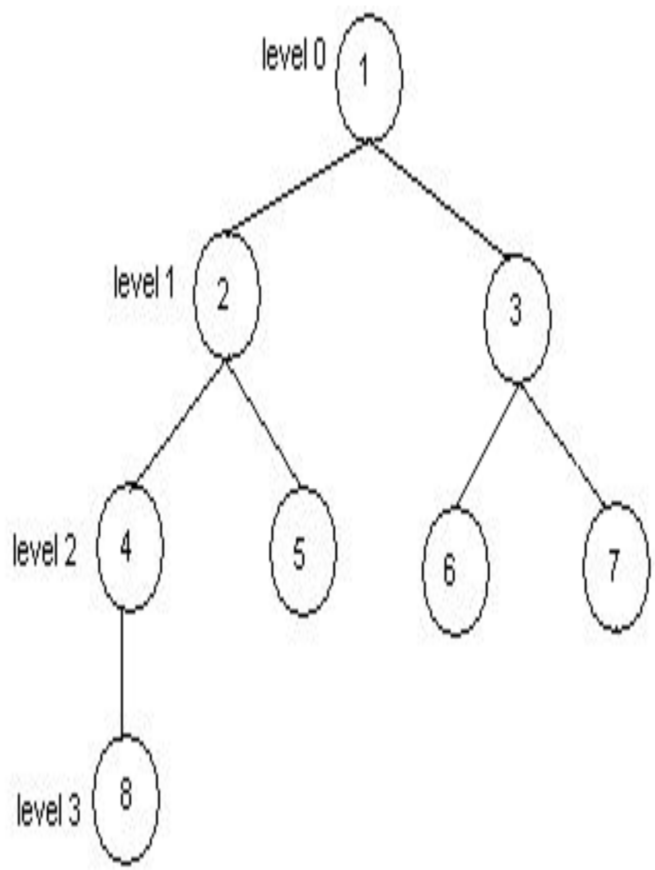


Figure: Copy T1 and T3.

Binary Tree

- *Edge:* A line from a node N of T to a successor is called is an edge.
- *Path:* A sequence of consecutive edges is called a path.
- *Branch:* A path from root node to a leaf node is called branch.
- *Level of Binary Tree:* Each node in a binary tree T is assigned a level number. The root R of T has level number 0 and every other node has level number which is one more than the level number of its parent.
- *Depth of Binary Tree:* Maximum number of nodes in a branch of T is the depth of T.

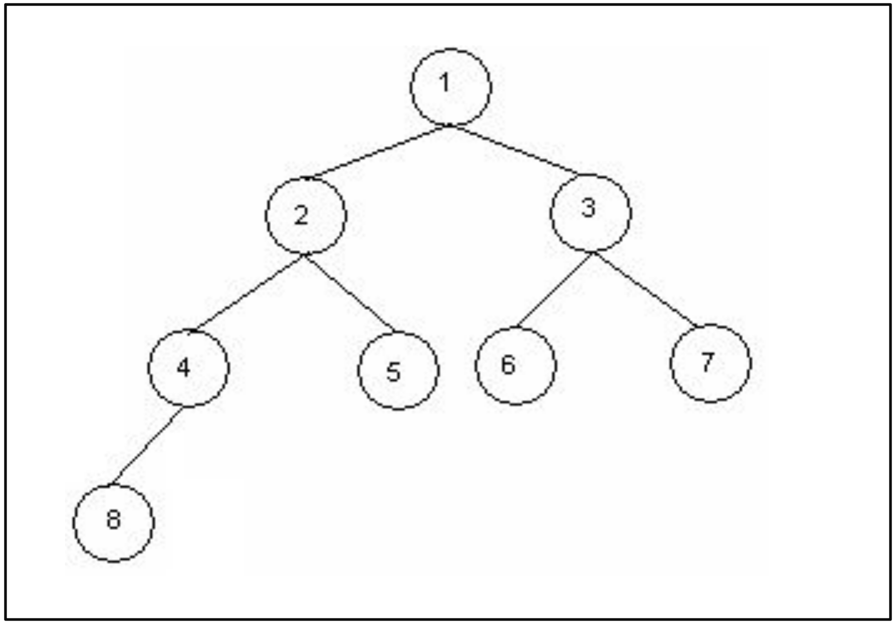
Binary Tree



Binary Tree: T
Edge: (1, 2), (3, 6) ....
Path: (1, 2, 4), (1, 3, 6)
Branch: (1, 2, 4, 8), (1, 2, 5), (1, 3, 6), (1, 3, 7)
Depth: 4

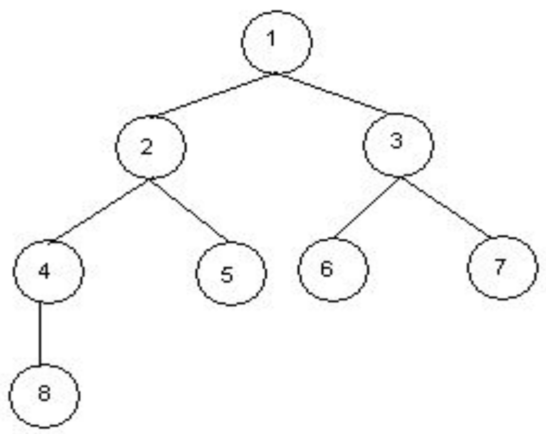
Complete Binary Trees

- A binary tree T is said to be complete if
  - all its level, except possibly the last, have the maximum number of possible nodes
  - all the nodes at the last level appear as far left as possible.
  - The depth D_n of the complete binary tree with n nodes $\lfloor \log_2 n + 1 \rfloor$

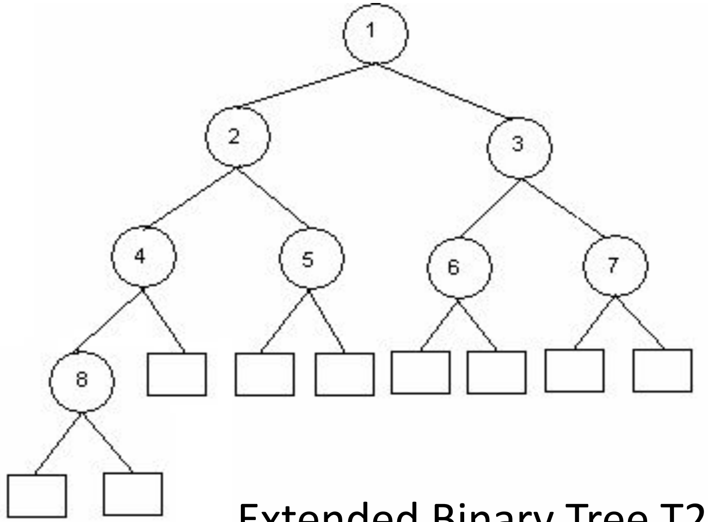


Extended Binary Tree (2-Tree)

- A binary tree T is said to be an extended binary tree if
  - each node N has either 0 or 2 children.
  - Nodes with 2 children are called internal nodes.
  - Nodes with 0 children are called external nodes.
  - Internal nodes are represented by circles and external nodes by squares.



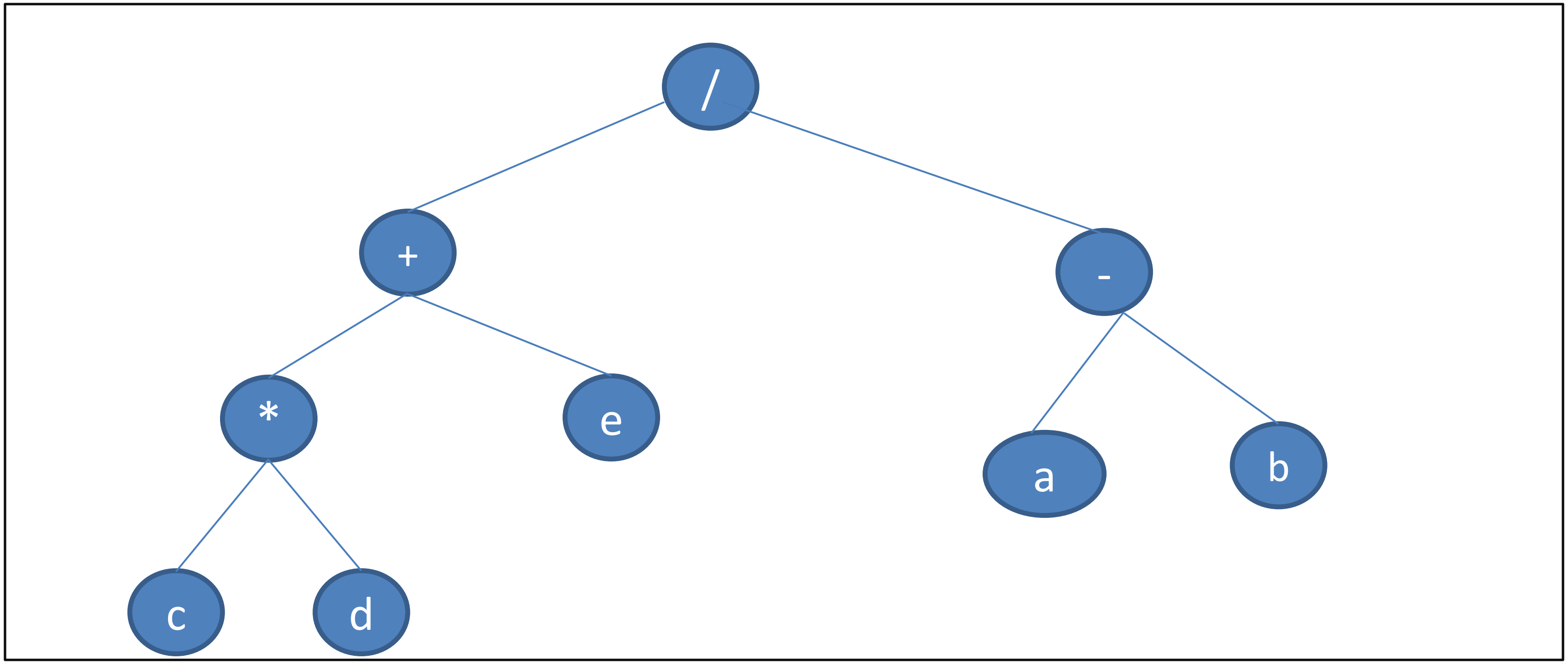
Binary Tree T1



Extended Binary Tree T2

Algebraic Expression as Binary Tree

- An algebraic expression E can be represented by means of binary tree T
- Example: ((c*d)+e)/(a-b)

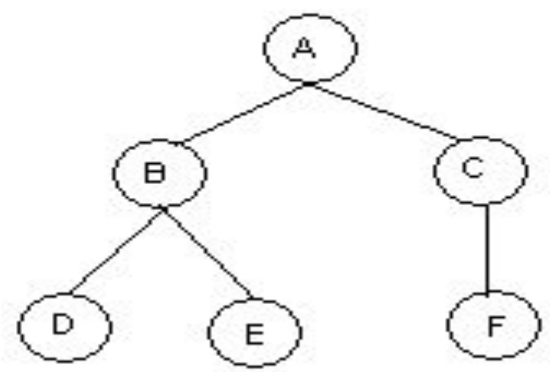


Representing Binary Tree in Memory

- Let T be a binary tree, then T can be represented in memory using two ways.
  - 1. Linked Representation
  - 2. Sequential Memory Representation/ Array representation

Linked Representation of Binary Tree

- Use Three parallel arrays Info, Left and Right and a pointer variable Root.
  - Info[K]: Contains data at node N.
  - Left[K]: Contains location of left child of N
  - Right[K]: Contains location of right child of N
  - Root: Contains location of Root

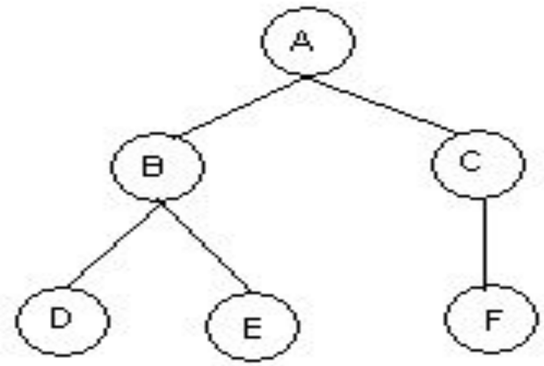


Root → 5

	Info	Left	Right
1	C	0	10
2	D	0	0
3			
4	E	0	0
5	A	7	1
6			
7	B	2	4
8			
9			
10	F	0	0

Sequential Representation of Binary Tree

- Use only a single liner array Tree.
  - Tree[1] represents the Root of T.
  - If node N is in Tree[K], then its left child is in Tree[2K]
  - If node N is in Tree[K], then its right child is in Tree[2K+1].

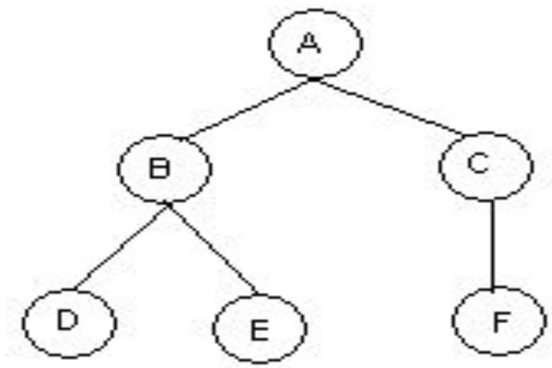


Tree =

1	2	3	4	5	6	7	8	9	10
A	B	C	D	E		F			

Sequential Representation of Binary Tree

- If a tree has depth d then it will require a array of maximum 2^{d+1}.
  - If T has depth 5
  - then it requires an array of 2^{5+1}=64 elements



$?=2^3$

<u>Traversing a Binary Tree</u>

Traversing Binary Tree

There are 3 ways of traversing a binary tree T having root R.

1. Preorder Traversing

- Steps:

(a) Process the root R

(b) Traverse the left subtree of R in preorder.

(c) Traverse the right subtree of R in preorder.

- Example:

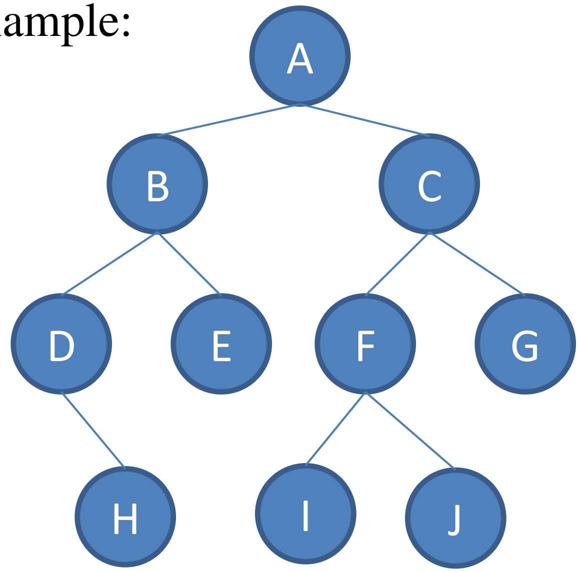


Figure: Binary Tree T

Preorder Traversal of T

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

2. Inorder Traversing

- Steps:

  (a) Traverse the left subtree of R in inorder.

  (b) Traverse the root R.

  (c) Traverse the right subtree of R in inorder.

- Example:

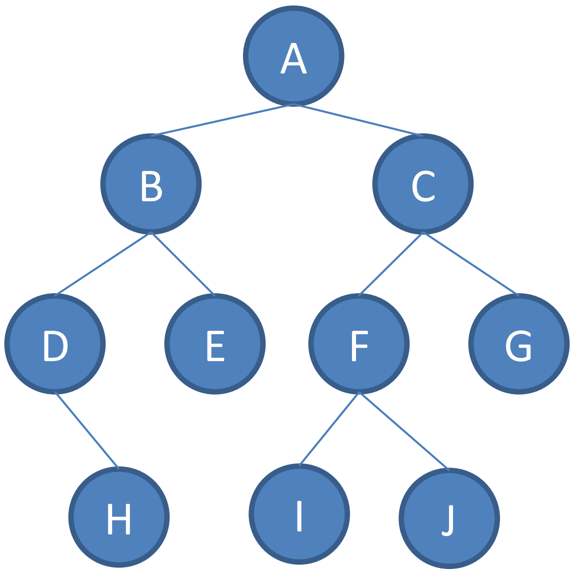


Figure: Binary Tree T

<u>Inorder Traversal of T</u>

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

3. Postorder Traversing

- Steps:

  (a) Traverse the left subtree of R in postorder.

  (b) Traverse the right subtree of R in postorder.

  (c) Traverse the root R.

- Example:

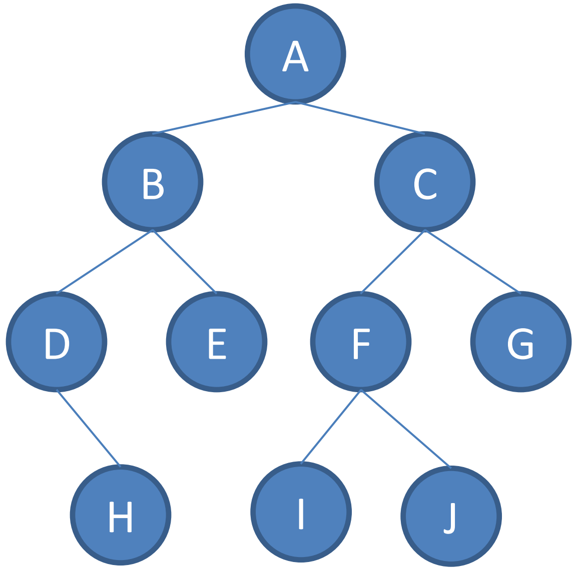


Figure: Binary Tree T

<u>Postorder Traversal of T</u>

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

End