CS-2001
DATA STRUCTURE

Dr. Hashim Yasin

National University of Computer and Emerging Sciences,

Faisalabad, Pakistan.

TREE

Tree

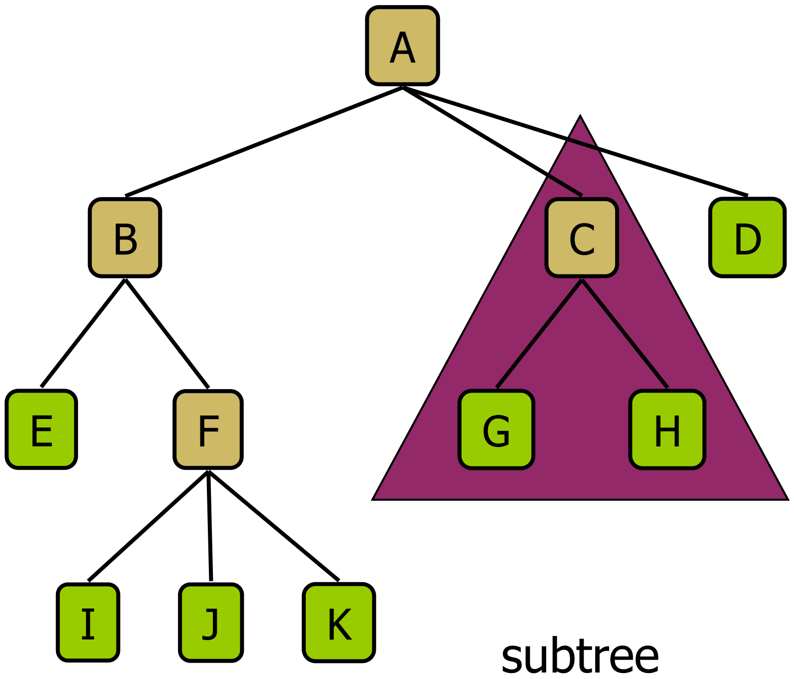
- A tree is a finite nonempty set of elements.
- It is an *abstract model of a hierarchical structure*.
- A tree consists of nodes with a parent-child relation. Edges are used for that purpose
  - Recursive data structure
  - Root -> sub trees (Left & right)
  - All node one incoming link
  - Many outgoing links
  - Total (n-1) links. Root has no incoming link

Applications

- File system
- Organizing data for quick search, insertion and deletion
- Tree is used for dictionary implementation
- Networking routing algorithms
- Organization charts
- Programming environments
- Family tree

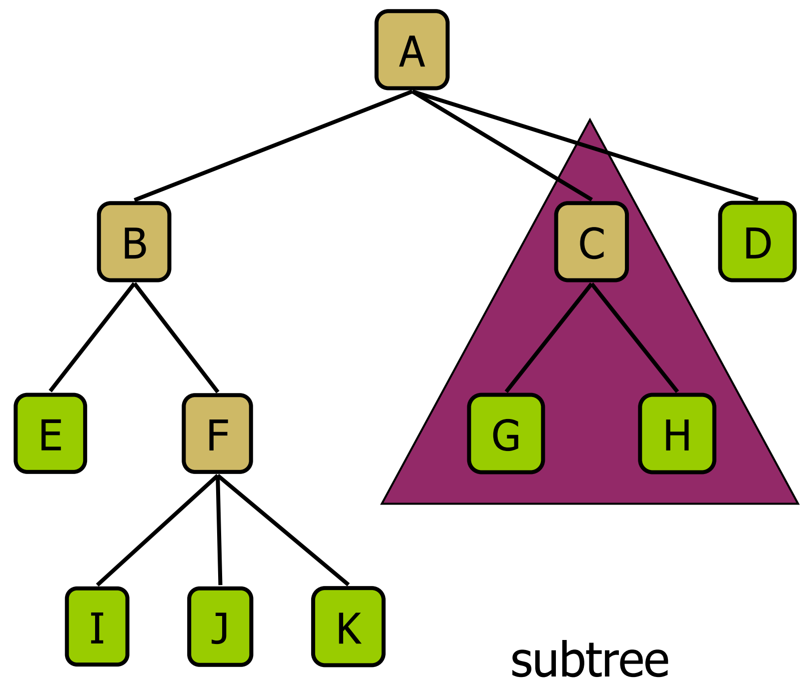
Tree … Terminologies

- **Root:** node without parent (A)
- **Siblings:** nodes share the same parent
- **Internal node:** node with at least one child (A, B, C, F)
- **External node (leaf):** node without children (E, I, J, K, G, H, D)



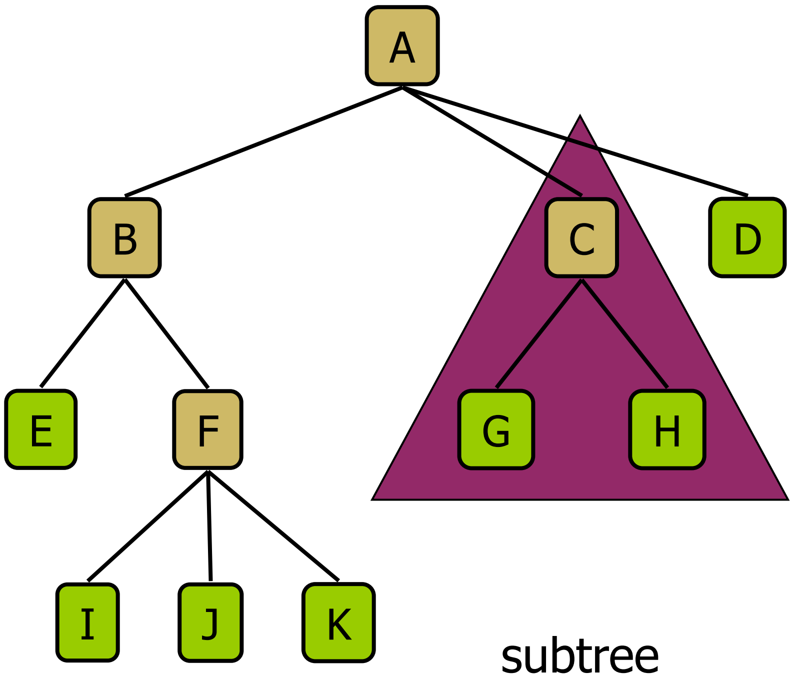
Tree … Terminologies

- **Ancestors** of a node: parent, grandparent, grand-grandparent, etc.
- **Descendant** of a node: child, grandchild, grand-grandchild, etc.
- **Height** of a tree:
  - maximum depth of any node (3), OR
  - the number of edges along the longest path from the node to a leaf



Tree … Terminologies

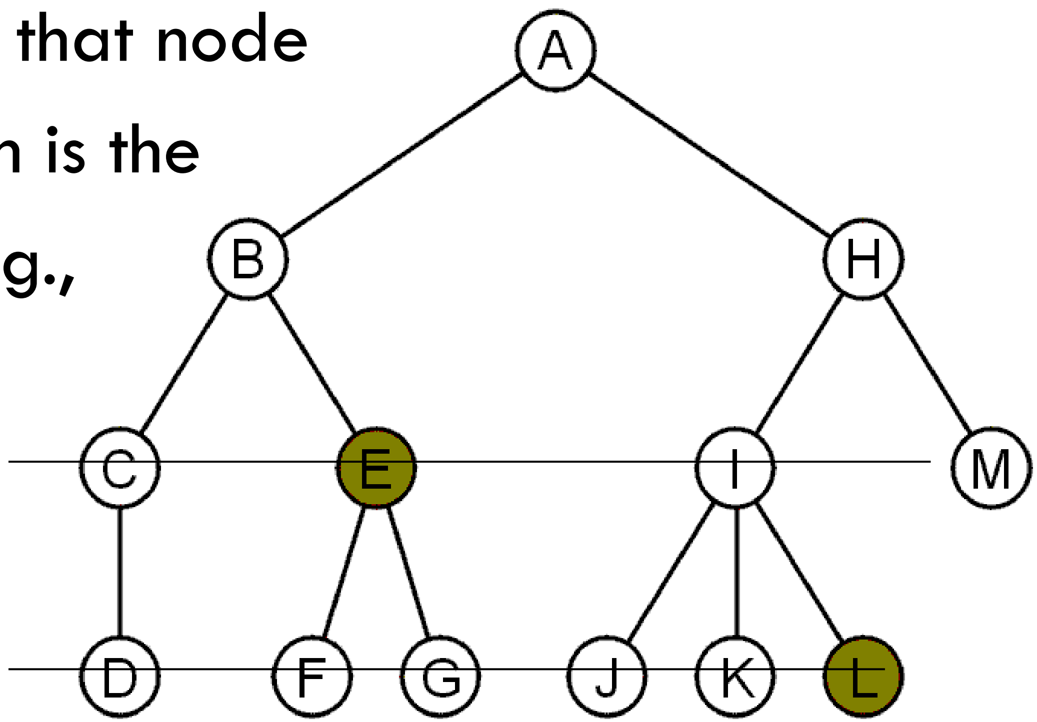
- **Depth** of a node: number of ancestors
- **Degree of a node:** the number of its children
- **Degree of a tree:** the maximum number of its node



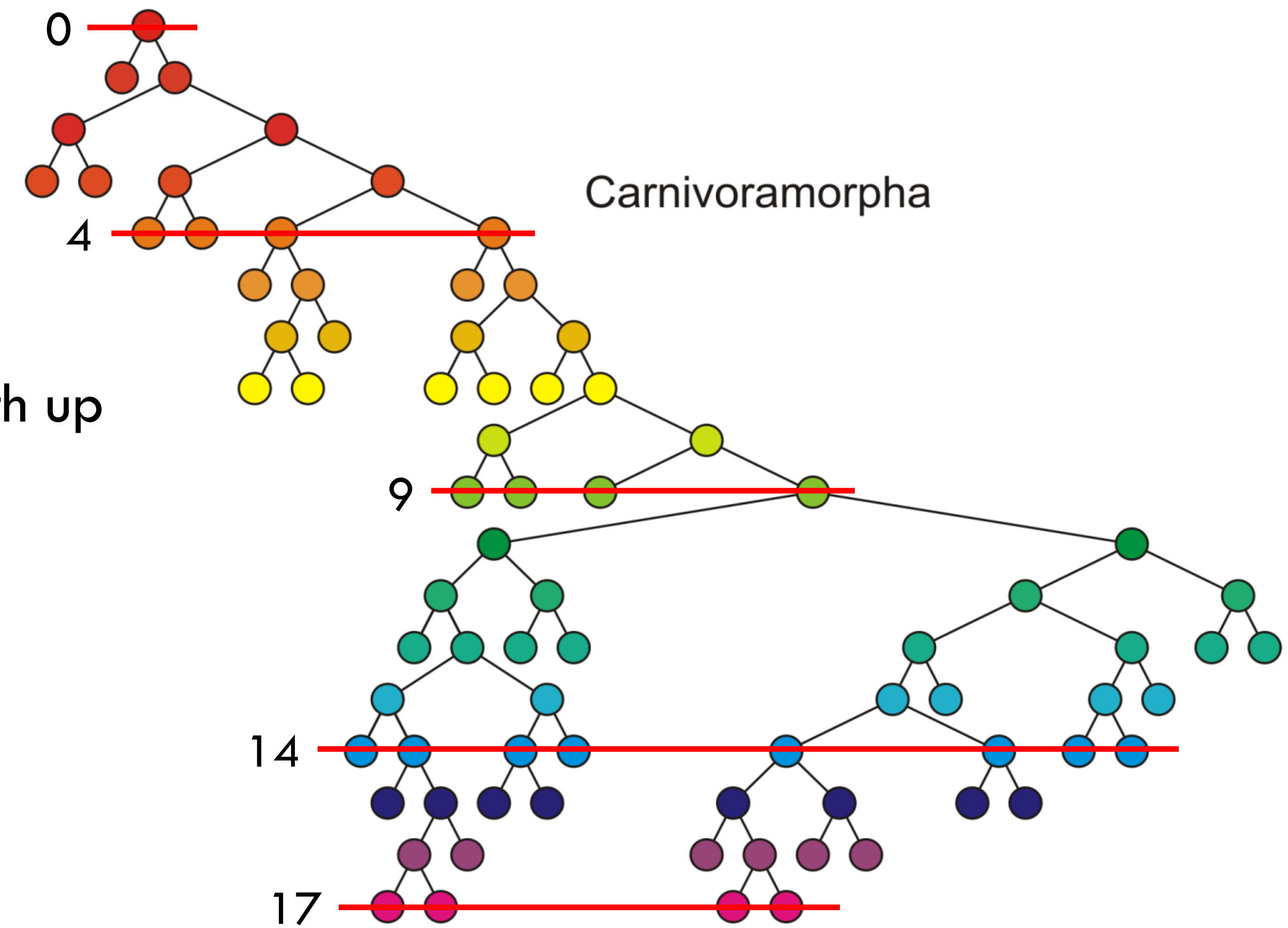
Tree … Terminologies

Depth of the node

- For each node in a tree, there exists a unique path from the root node to that node
- The length of this path is the **depth of the node**, e.g.,
  - E has depth 2
  - L has depth 3



Tree … Terminologies



Nodes of depth up to 17

Wesley-Hunt, G. D.; Flynn, J. J. “Phylogeny of the Carnivora: basal relationships among the Carnivoramorphans, and assessment of the position of ‘Miacoidea’

Tree … Terminologies

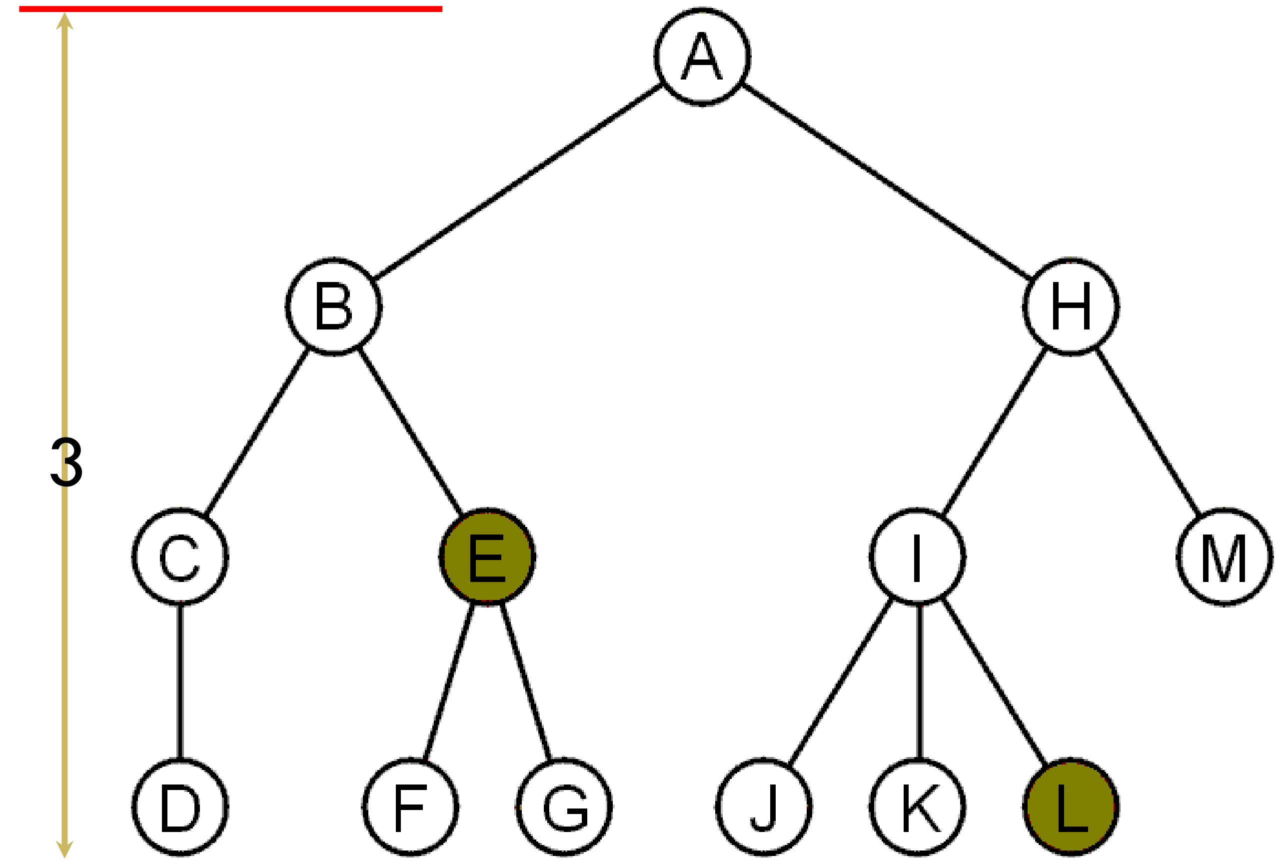
- The **<u>height of a tree</u>** is defined as the maximum depth of any node within the tree

- The height of a tree with one node is 0
  - Just the root node

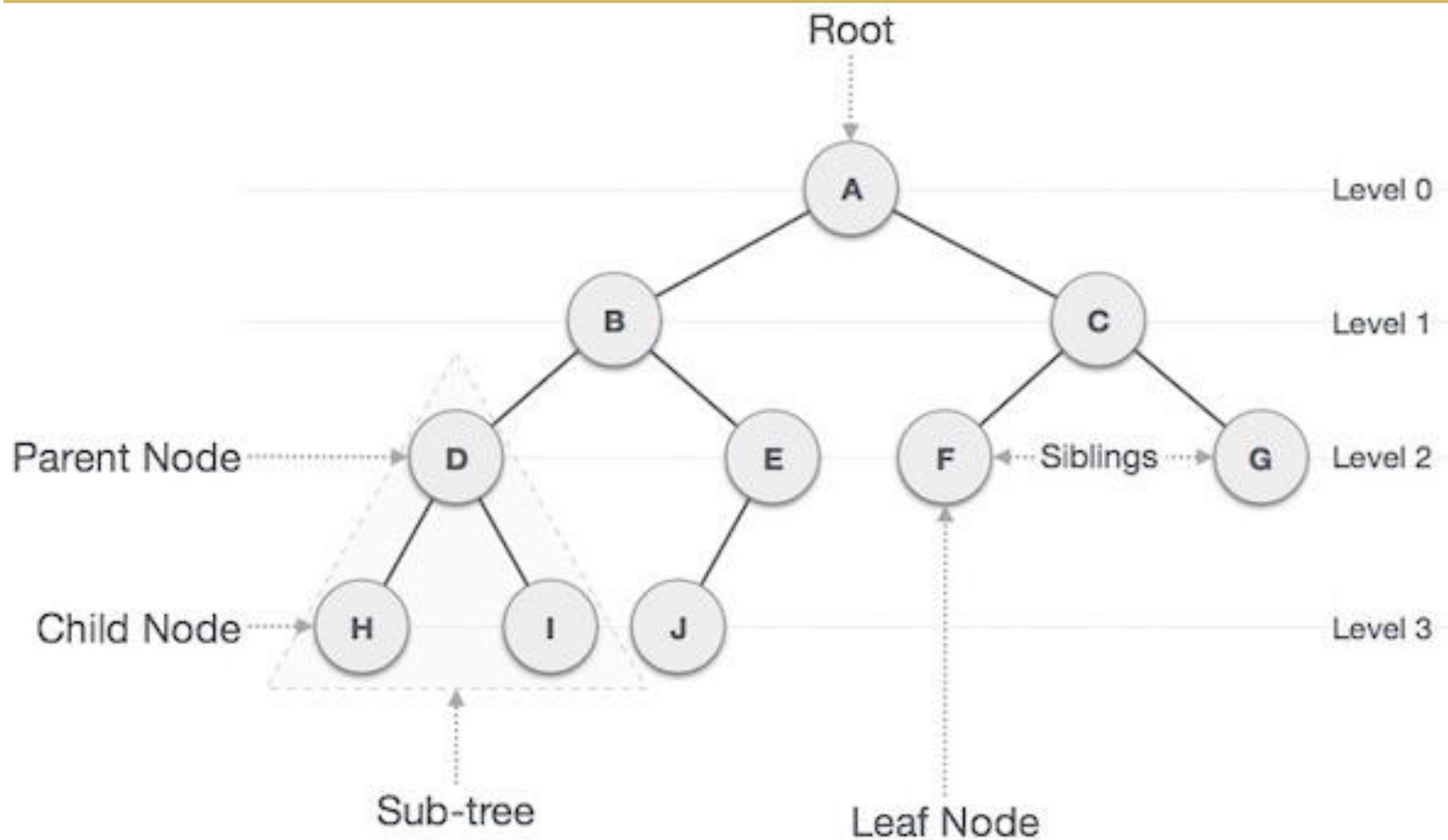
- For convenience, we define the height of the empty tree to be –1

Tree … Terminologies

Height of the tree is 3



Tree … Terminologies



TREE TRAVERSAL

Tree Traversals

- Trees can be traversed in different ways.
- Following are the generally used ways for traversing trees.

Inorder

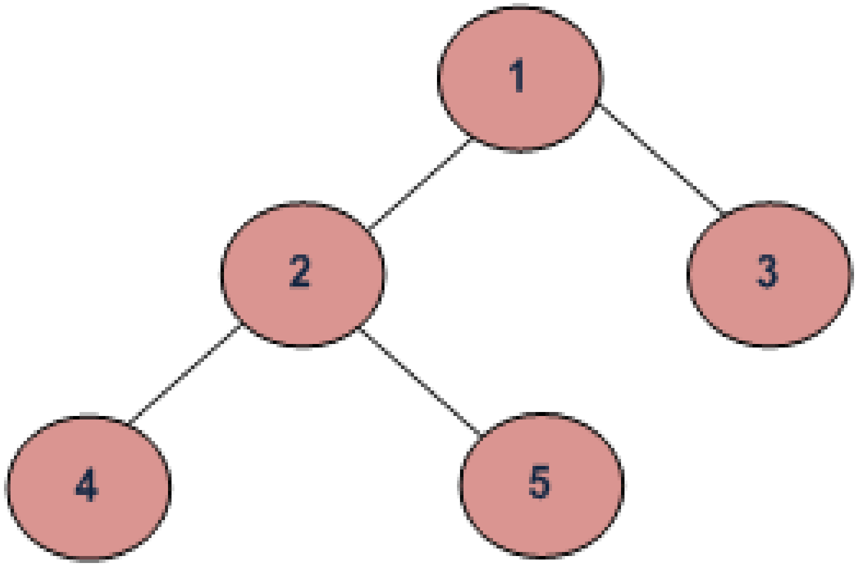
- (Left, Root, Right) : 4 2 5 1 3

Preorder

- (Root, Left, Right) : 1 2 4 5 3

Postorder

- (Left, Right, Root) : 4 5 2 3 1



Tree Traversals

- **Algorithm Inorder(tree)**

  1. Traverse the left subtree, i.e., call Inorder(left-subtree)

  2. Visit the root.

  3. Traverse the right subtree, i.e., call Inorder(right-subtree)

Tree Traversals

- **Algorithm Preorder(tree)**
  1. Visit the root.
  2. Traverse the left subtree, i.e., call Preorder(left-subtree)
  3. Traverse the right subtree, i.e., call Preorder(right-subtree)

Tree Traversals

- **Algorithm Postorder(tree)**

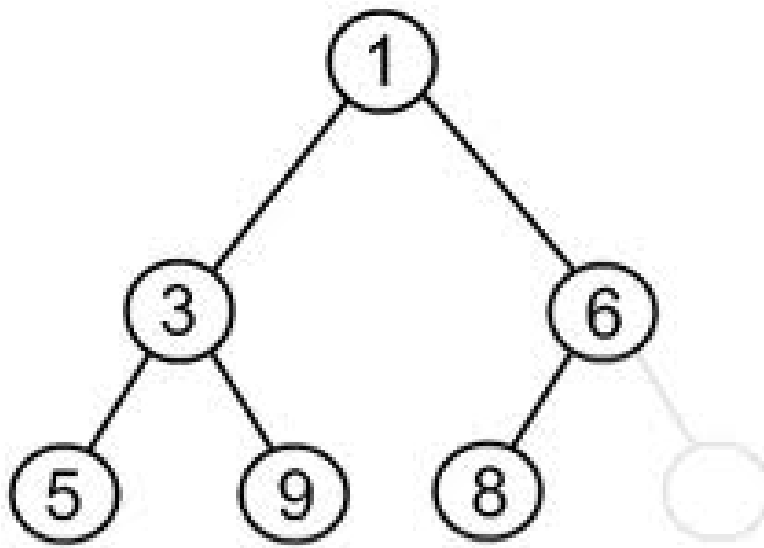
    1. Traverse the left subtree, i.e., call Postorder(left-subtree)

    2. Traverse the right subtree, i.e., call Postorder(right-subtree)

    3. Visit the root.

Tree Traversals … Example

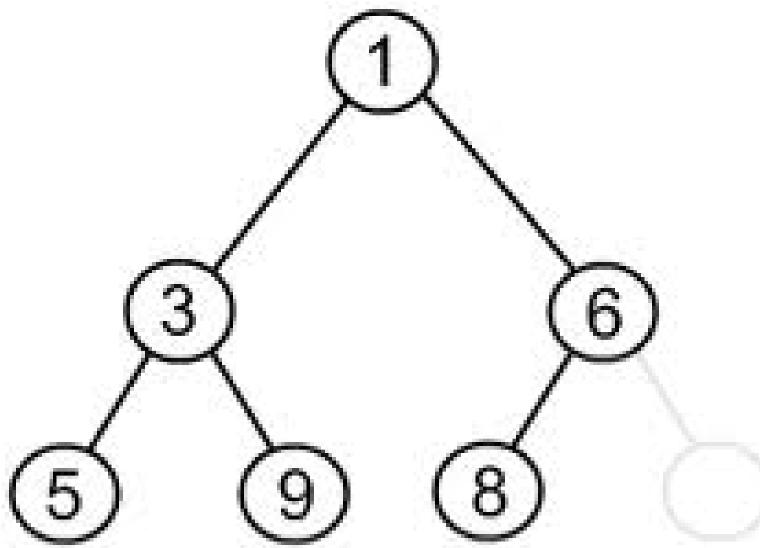
In-Order(Left-Root-Right)



5, 3, 9, 1, 8, 6

Tree Traversals … Example

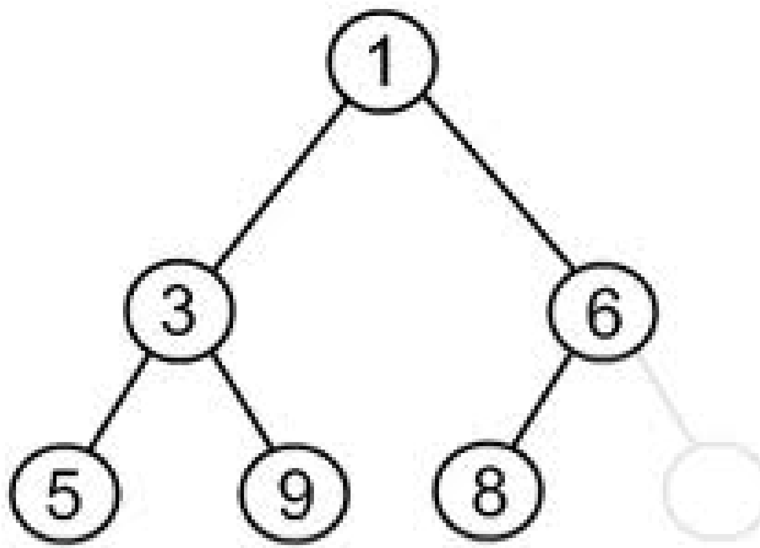
Pre-Order(Root-Left-Right)



1, 3, 5, 9, 6, 8

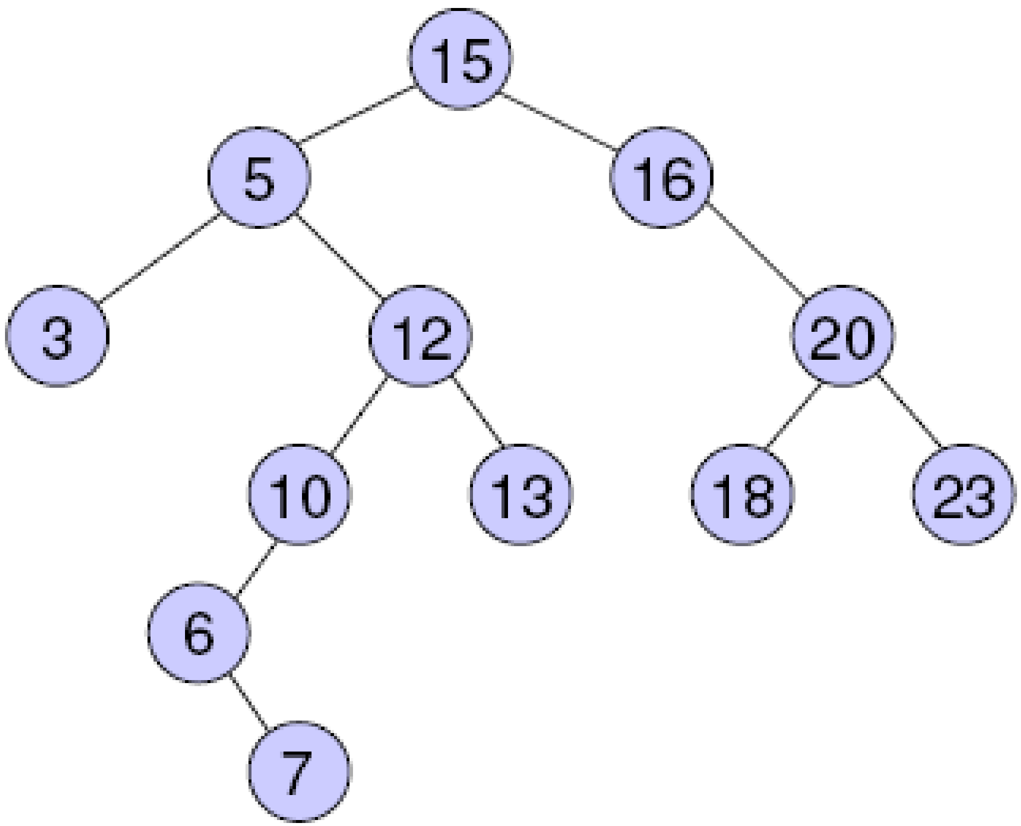
Tree Traversals … Example

Post-Order(Left-Right-Root)



5, 9, 3, 8, 6, 1

Tree Traversal another Example



Tree Traversal another Example

- **In-order: (left, root, right)**

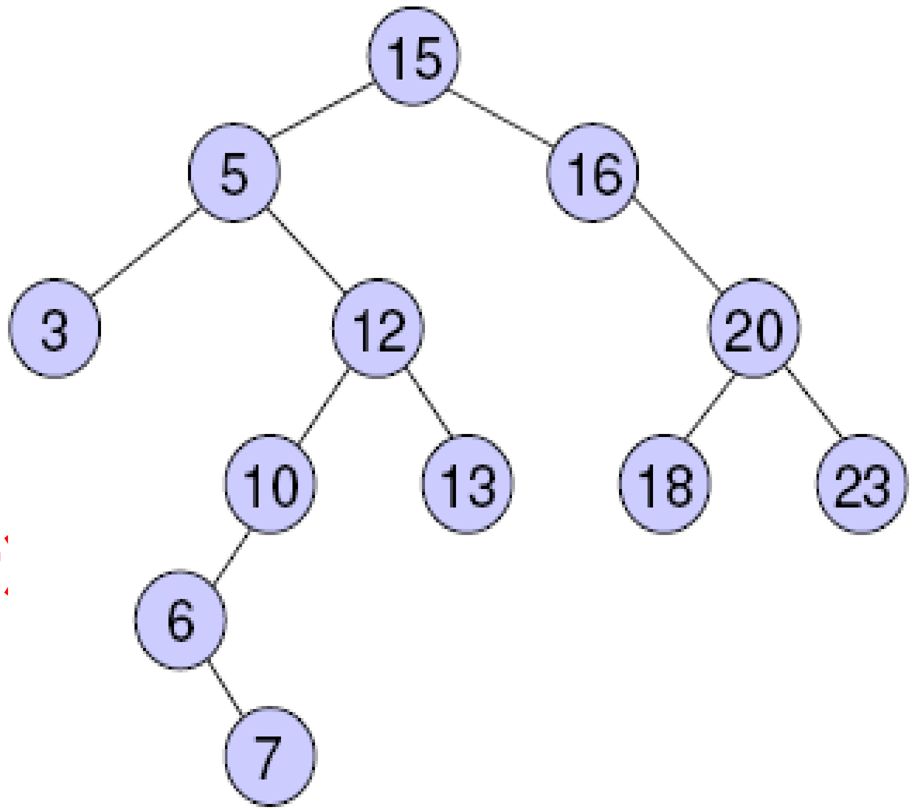
  3, 5, 6, 7, 10, 12, 13,
  15, 16, 18, 20, 23

- **Pre-order: (root, left, right)**

  15, 5, 3, 12, 10, 6, 7,
  13, 16, 20, 18, 23

- **Post-order: (left, right, root)**

  3, 7, 6, 10, 13, 12, 5,
  18, 23, 20, 16, 15



Reading Materials

- Nell Dale Chapter#8
- Schaum's Outlines Chapter#7
- D. S. Malik Chapter#11