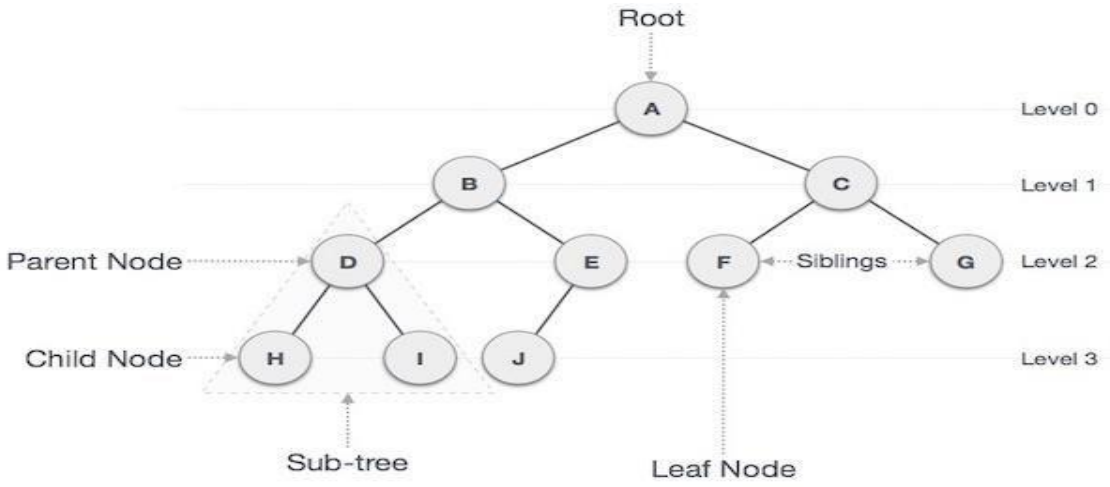
CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

TREE
(CAT-201)

Design By:
Ms. Gurpreet kaur dhiman
Ms.Mandeep kaur
Chandigarh University-Gharuan

TREE(binary tree)

- Tree represents the nodes connected by edges.
- Binary Tree is a special data structure used for data storage purposes. A binary tree has a special condition that each node can have a maximum of two children. A binary tree has the benefits of both an ordered array and a linked list as search is as quick as in a sorted array and insertion or deletion operation are as fast as in linked list.



https://www.tutorialspoint.com/data_structures_algorithms/tree_.htm

IMPORTANT TERMS IN TREE

Following are the important terms with respect to tree.

- **Path** − Path refers to the sequence of nodes along the edges of a tree.
- **Root** − The node at the top of the tree is called root. There is only one root per tree and one path from the root node to any node.
- **Parent** − Any node except the root node has one edge upward to a node called parent.
- **Child** − The node below a given node connected by its edge downward is called its child node.
- **Leaf** − The node which does not have any child node is called the leaf node.
- **Subtree** − Subtree represents the descendants of a node.

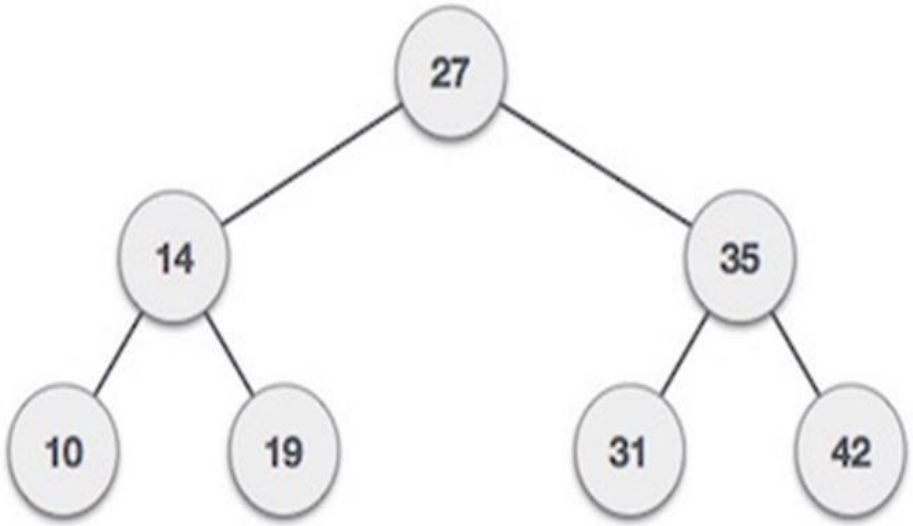
CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

IMPORTANT TERMS IN TREE

- **Visiting** − Visiting refers to checking the value of a node when control is on the node.
- **Traversing** − Traversing means passing through nodes in a specific order.
- **Levels** − Level of a node represents the generation of a node. If the root node is at level 0, then its next child node is at level 1, its grandchild is at level 2, and so on.
- **keys** − Key represents a value of a node based on which a search operation is to be carried out for a node.

Binary search tree

Binary Search tree exhibits a special behavior. A node's left child must have a value less than its parent's value and the node's right child must have a value greater than its parent value.



Reference:
https://www.tutorialspoint.com/data_structures_algorithms/tree_binary.htm

TREE TRAVERSAL

CU
CHANDIGARH UNIVERSITY

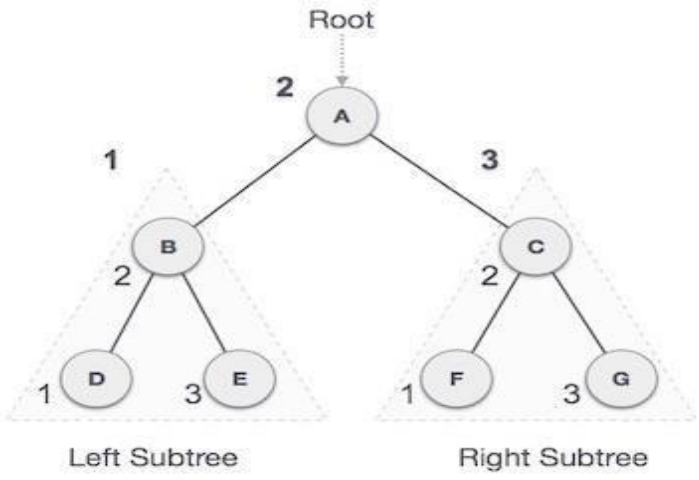
Traversal is a process to visit all the nodes of a tree and may print their values too. Because, all nodes are connected via edges (links) we always start from the root (head) node. That is, we cannot randomly access a node in a tree.

There are three ways which we use to traverse a tree −

- In-order Traversal
- Pre-order Traversal
- Post-order Traversal

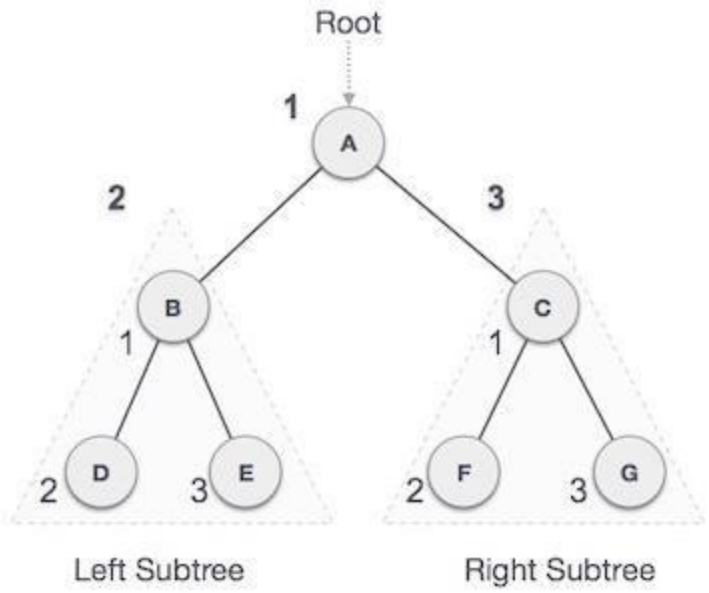
IN-ORDER TRAVERSAL

- In-order Traversal
- In this traversal method, the left sub-tree is visited first, then the root and later the right sub-tree. We should always remember that every node may represent a sub-tree itself.
- If a binary tree is traversed **in-order**, the output will produce sorted key values in an ascending order.
- We start from **A**, and following in-order traversal, we move to its left sub-tree **B**. **B** is also traversed in-order. The process goes on until all the nodes are visited. The output of in-order traversal of this tree will be −
- $D \rightarrow B \rightarrow E \rightarrow A \rightarrow F \rightarrow C \rightarrow G$



PRE-ORDER TRAVERSAL

- In this traversal method, the root node is visited first, then the left subtree and finally the right subtree.
- We start from **A**, and following pre-order traversal, we first visit **A** itself and then move to its left subtree **B**. **B** is also traversed pre-order. The process goes on until all the nodes are visited. The output of pre-order traversal of this tree will be −
- $A \rightarrow B \rightarrow D \rightarrow E \rightarrow C \rightarrow F \rightarrow G$

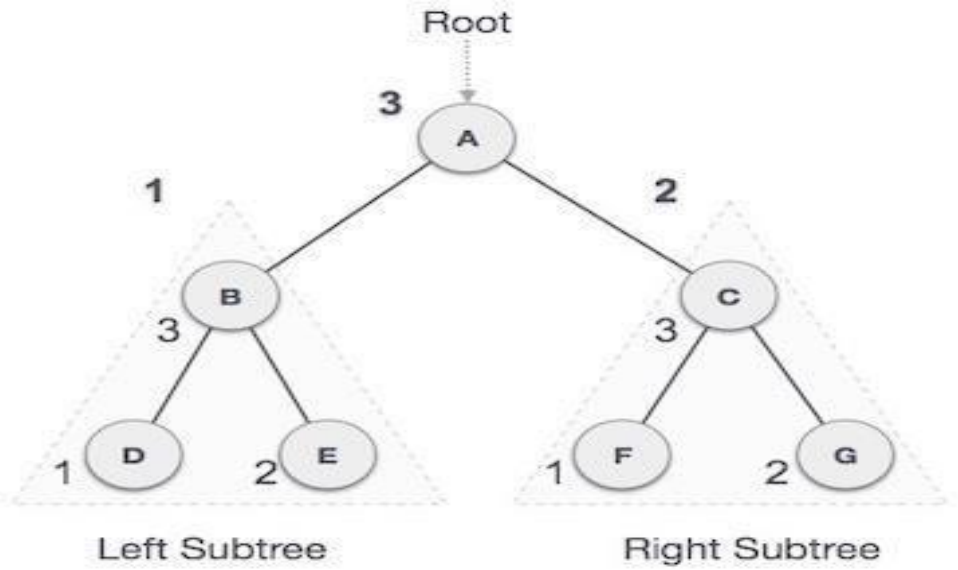


POST-ORDER TRAVERSAL

In this traversal method, the root node is visited last, hence the name. First we traverse the left sub-tree, then the right sub-tree and finally the root node.

We start from **A**, and following pre-order traversal, we first visit the left sub-tree **B**. **B** is also traversed post-order. The process goes on until all the nodes are visited. The output of post-order traversal of this tree will be −

$$D \rightarrow E \rightarrow B \rightarrow F \rightarrow G \rightarrow C \rightarrow A$$



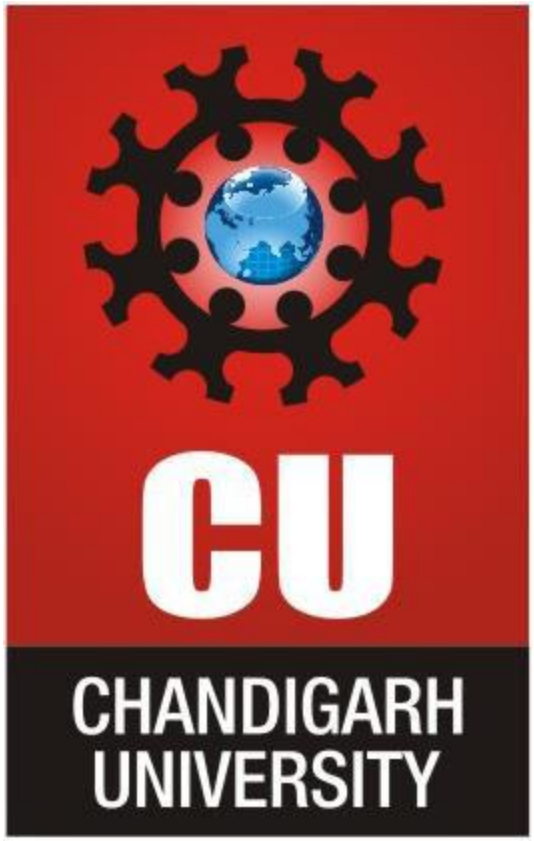
https://www.tutorialspoint.com/data_structures_algorithms/tree_traversal.htm

FAQ

- How tree height is measured?
- Differentiate binary and binary search tree?
- How traversing is performed in post order?

Bibliography

- Seymour Lipschutz, Schaum's Outlines Series Data structures TMH
- Introduction to Data Structures Applications, Trembley&Soreson, Second Edition, Pearson Education
- www.geeksforgeeks.org/analysis-of-algorithms-set-3asymptotic-notations/
- www.tutorialspoint.com/data_structures_algorithms/asymptotic_analysis.htm



Thank You