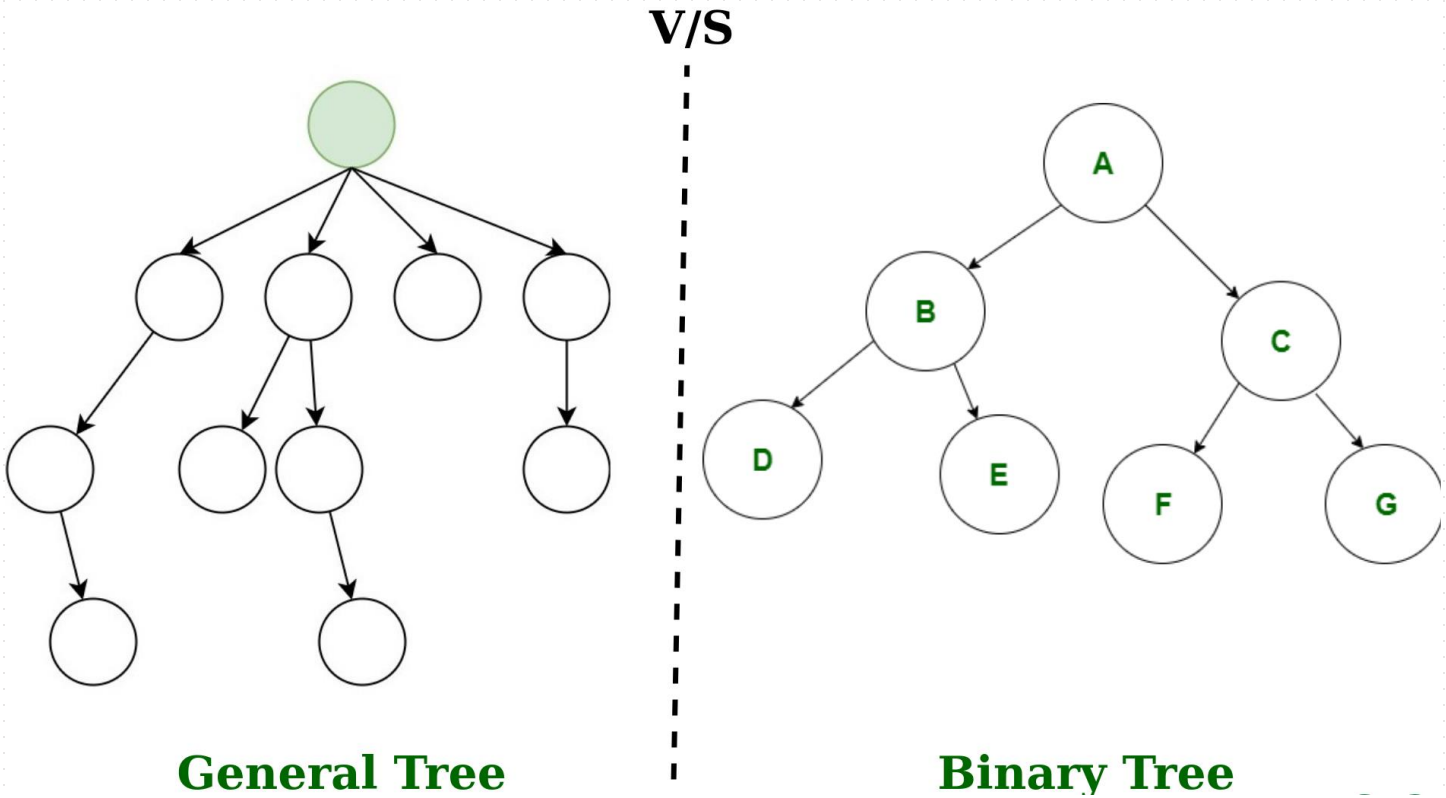
COMP-352 Tutorial #6

DS: Trees and Binary Tree

- **Tree** (General) data structure is **a hierarchical (non-linear)** data structure. It is a collection of nodes that are connected by edges and has a hierarchical relationship between the nodes.
- **Binary Tree**: Each node can have at most 2 child nodes.

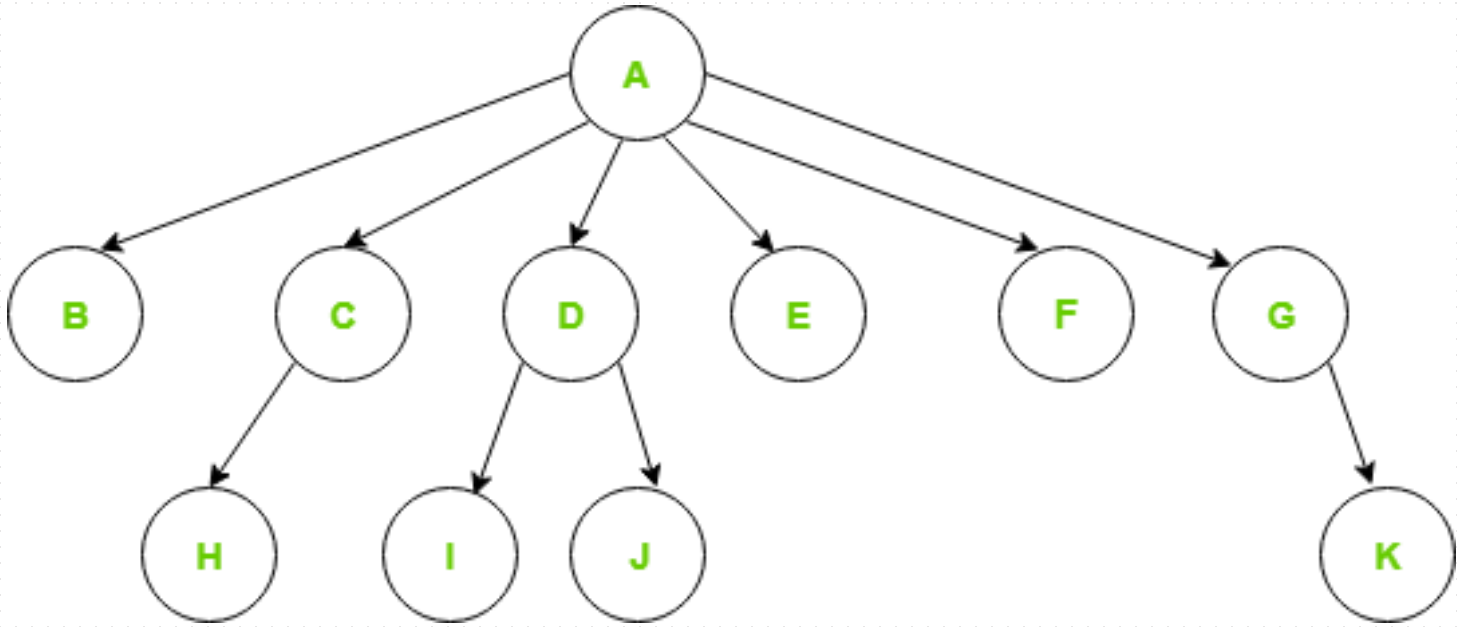


Source: https://www.geeksforgeeks.org/difference-between-general-tree-and-binary-tree/

Generic Trees

Definition: Generic trees are a collection of nodes where each node is a data structure that consists of records and a list of references to its children (duplicate references are not allowed).

- Unlike linked-list, each node can have multiple pointers to its children.
- The very first (top) node is called **root.**



Binary Tree

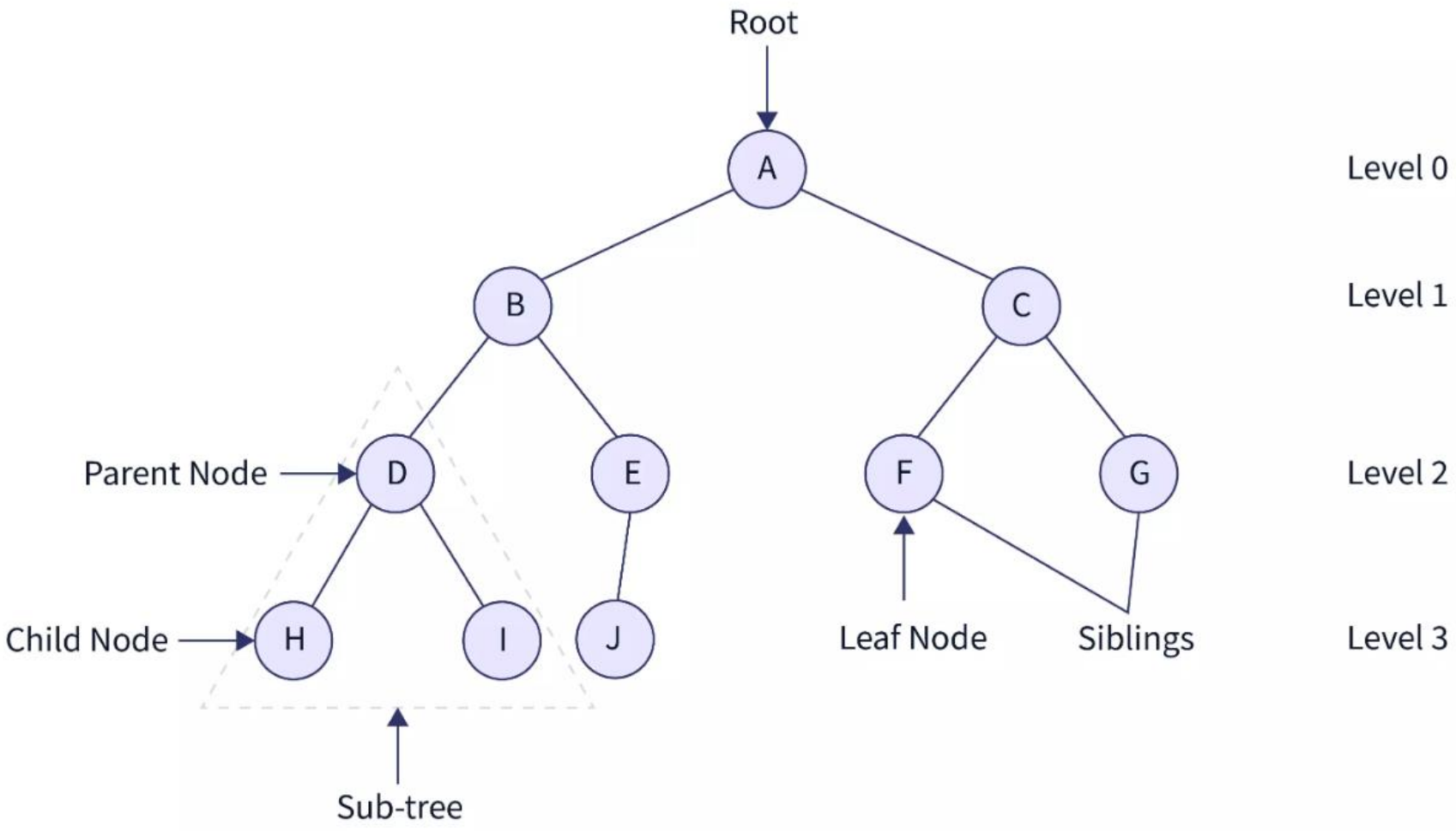
Definition: A tree data structure where each node has at most 2 children

- Thus, each node contains data, and left & right child pointers
- **Binary Trees** are a form of **Generic Trees**

Operations on a Binary tree:

- `Insert()`: Inserting a node at a specific place
- `Delete()`: Removing a certain node
- `Search()`: Search for an element
- `Traverse()`: Traversing the tree in certain order
- `Size()`: returning the size of the tree

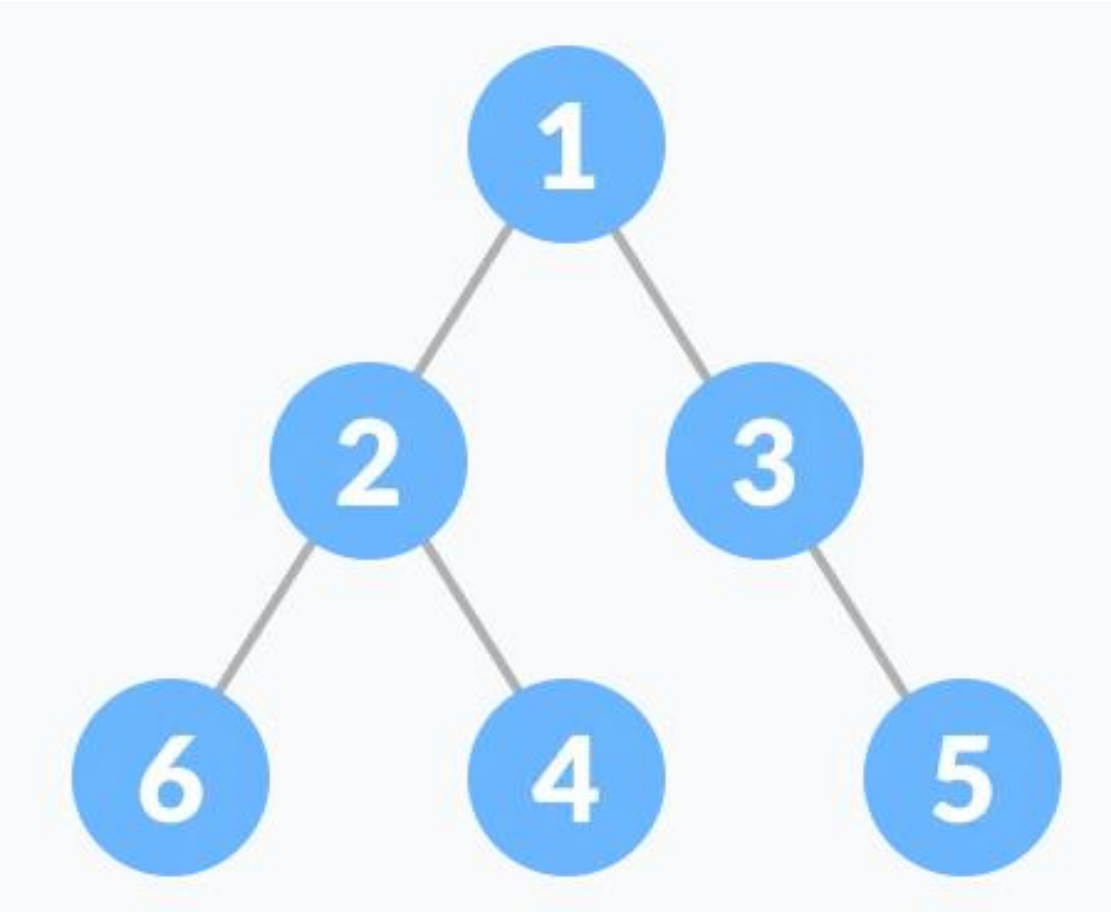
Binary Tree: A look



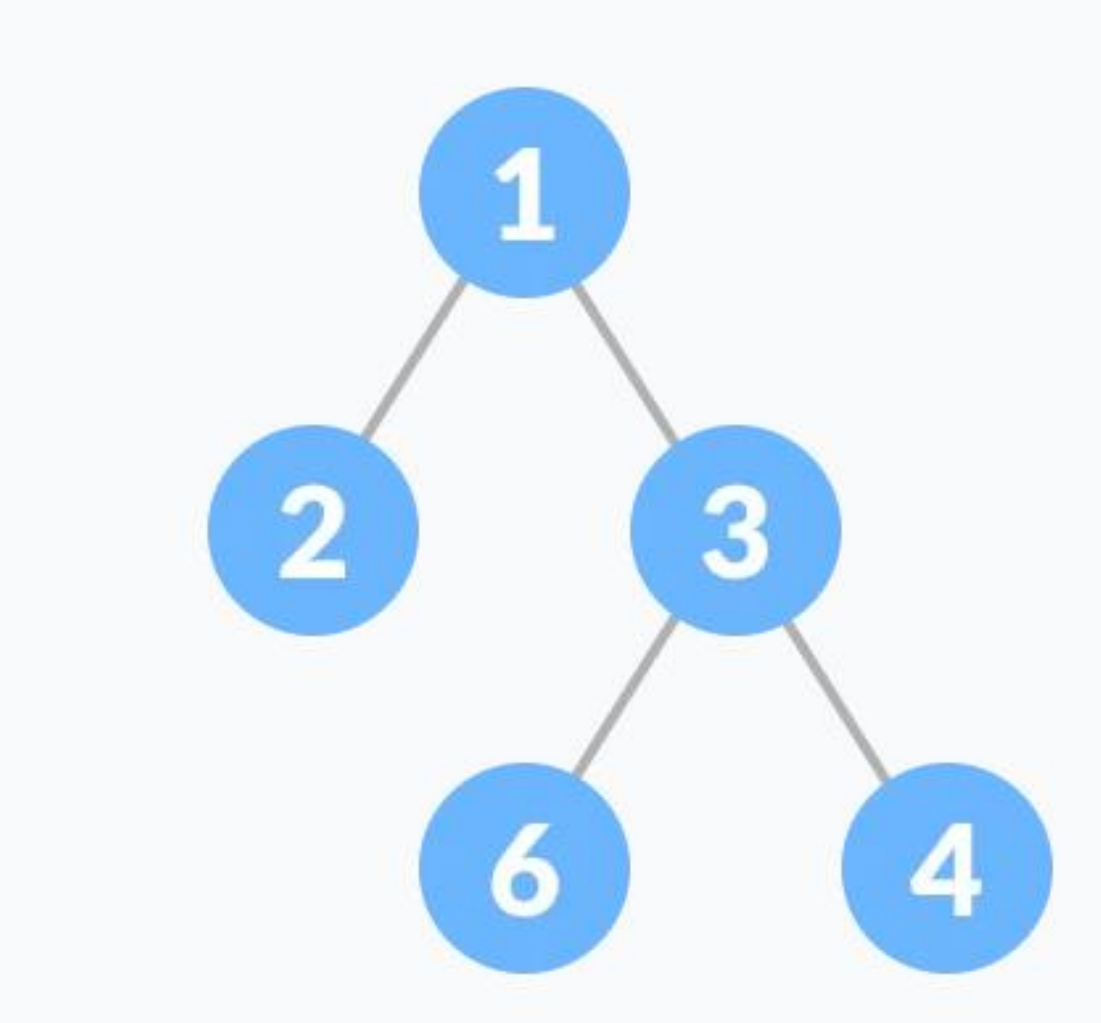
Binary Tree: Full, Complete & Balanced

- **Full Binary Tree:** every parent node/internal node has either two or no children
- **Complete Binary Tree:** similar to a full binary tree but:
  - All the leaf elements must lean towards the left.
  - The last leaf element might not have a right sibling *i.e., a complete binary tree doesn't have to be a full binary tree.*
- **Balanced Tree:** a binary tree in which the height of the left and right subtree of any node differ by not more than 1:
  - Difference between the left and the right subtree for any node is not more than one
  - The left subtree and right subtree are balanced

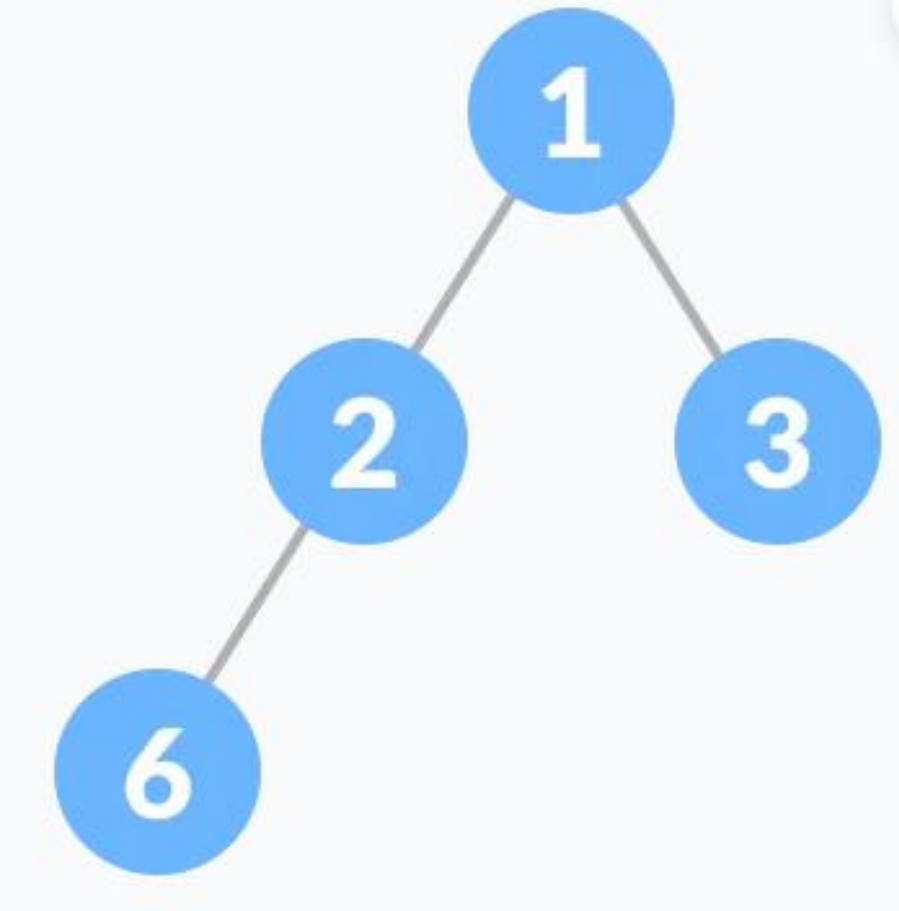
Tree Check: Is this tree Full or Complete?



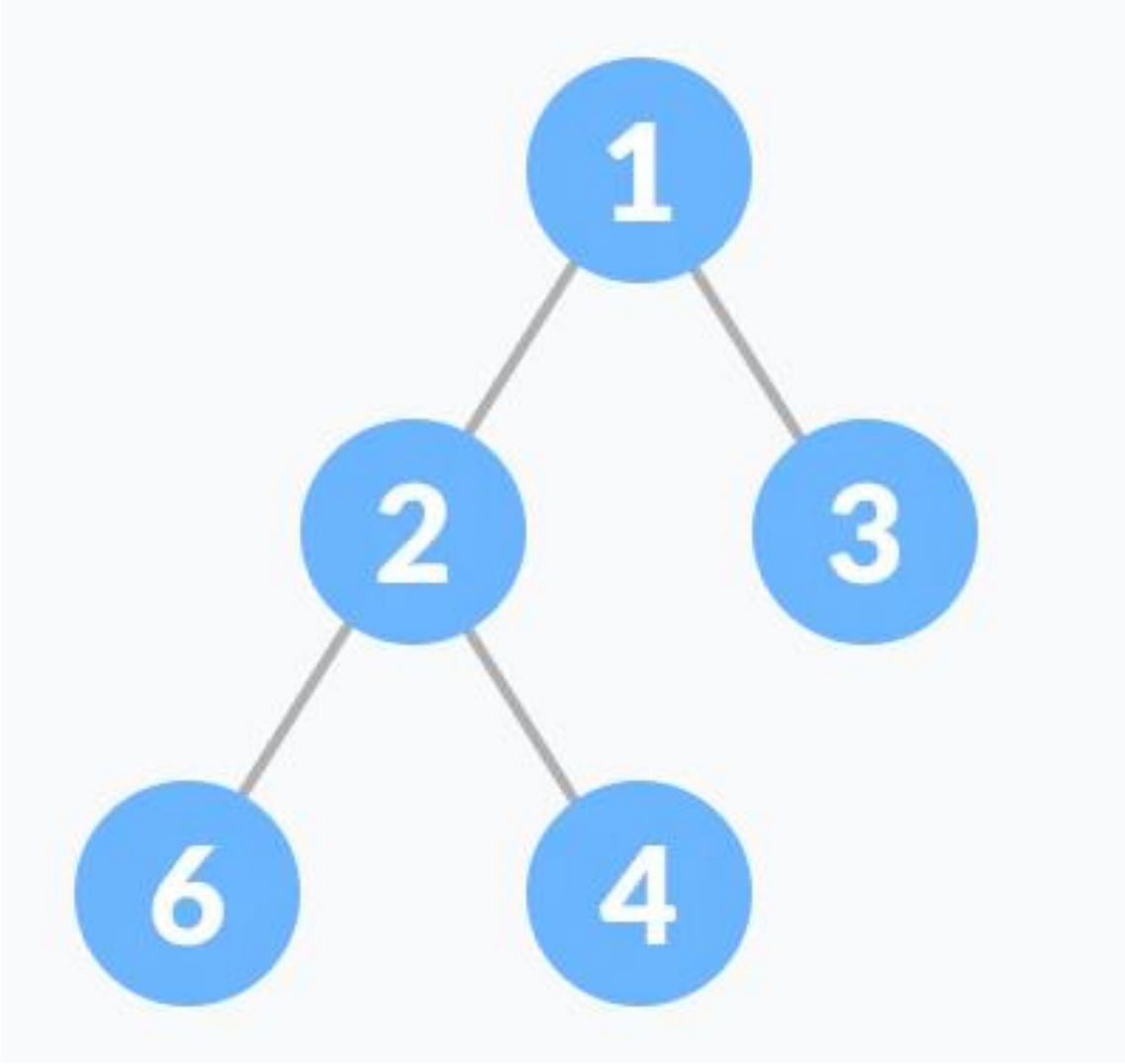
Tree Check: Is this tree Full or Complete?



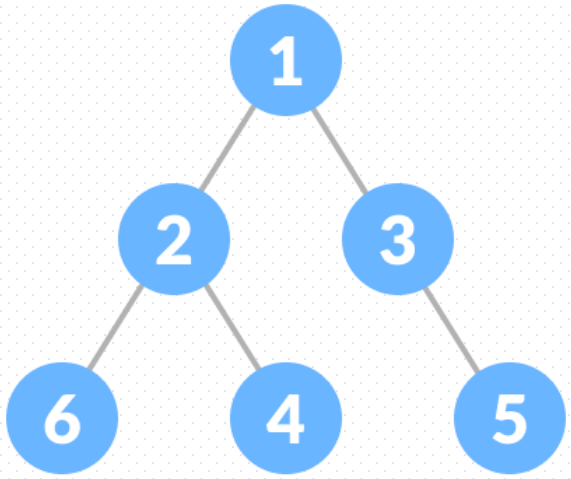
Tree Check: Is this tree Full or Complete?



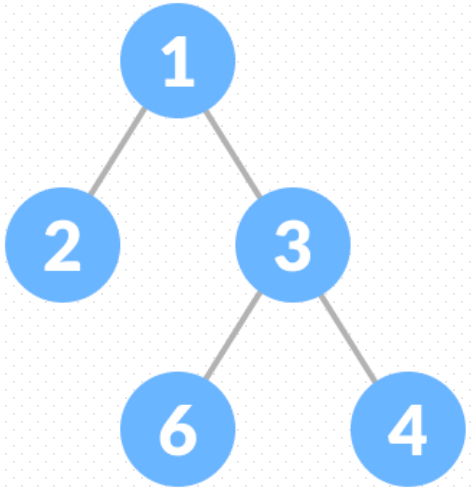
Tree Check: Is this tree Full or Complete?



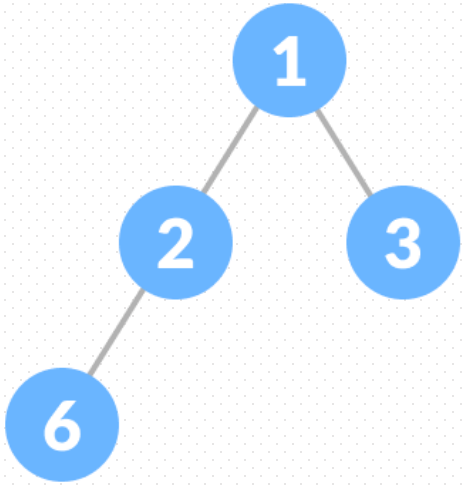
Tree Check: Is this tree Full or Complete?



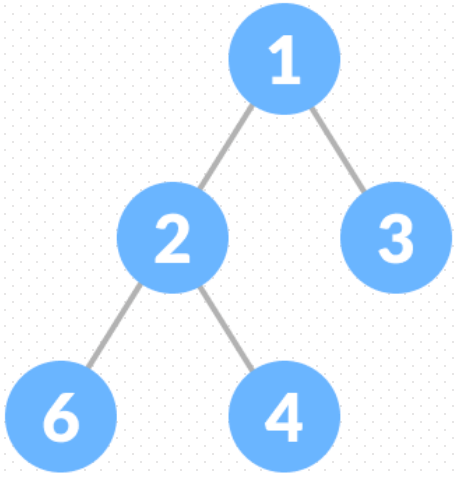
✗ Full Binary Tree
✗ Complete Binary Tree



✓ Full Binary Tree
✗ Complete Binary Tree



✗ Full Binary Tree
✓ Complete Binary Tree



✓ Full Binary Tree
✓ Complete Binary Tree

Source: https://www.programiz.com/dsa/complete-binary-tree

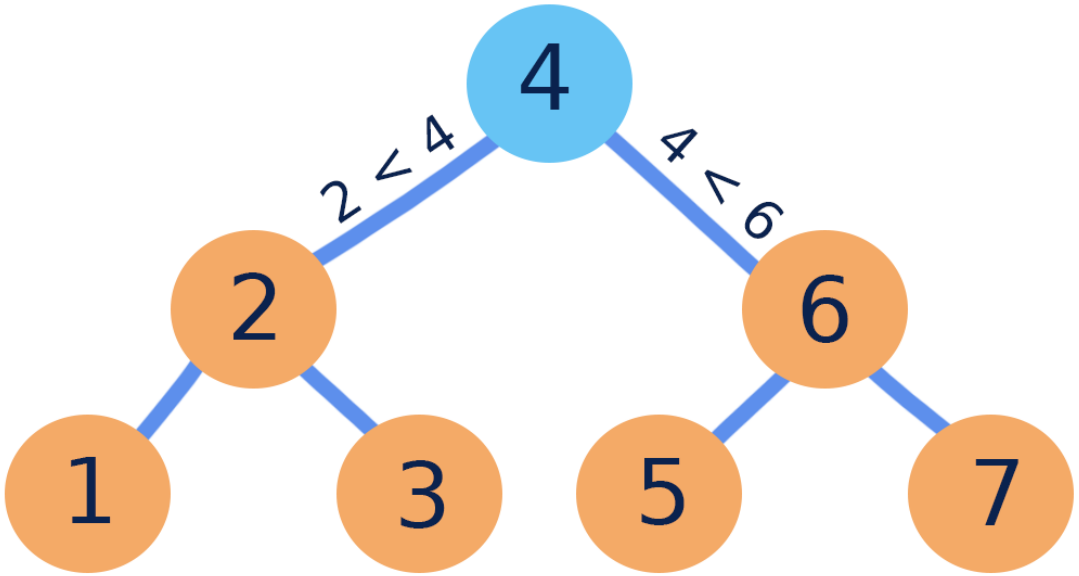
Tree Traversing

Traversing a Tree: travel across or through the tree

- **InOrder Traversing:**
  1. First, visit all the nodes in the left subtree
  2. Then the root node
  3. Visit all the nodes in the right subtree
- **PreOrder Traversing:**
  1. Visit root node
  2. Visit all the nodes in the left subtree
  3. Visit all the nodes in the right subtree
- **PostOrder Traversing:**
  1. Visit all the nodes in the left subtree
  2. Visit all the nodes in the right subtree
  3. Visit the root node

BST: Binary Search Tree

Definition: a node-based binary tree data structure in which for any given node the value in the node is bigger than the value in any node from the left sub-tree and smaller than any node in the right sub-tree.



In Order Traversal: 1 2 3 4 5 6 7

Exercise: Check if BST

TASK: Implement an algorithm that determines whether a binary tree is a BST or not.

- A Boolean value is to be returned as part of the check.
- Use the implementation of binaryTree from the GitHub.
- HINTS:
  - Use one of the traversal algos to check for BST
  - **OR** you can check recursively by comparing left and right

THANK YOU