CS062

DATA STRUCTURES AND ADVANCED PROGRAMMING

22: Binary Trees

Alexandra Papoutsaki
LECTURES

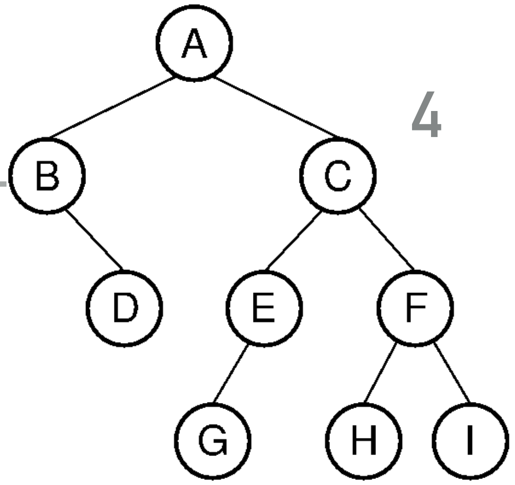
Mark Kampe
LABS

22: Binary Trees

- Binary Trees
- Tree traversals

Trees in CS

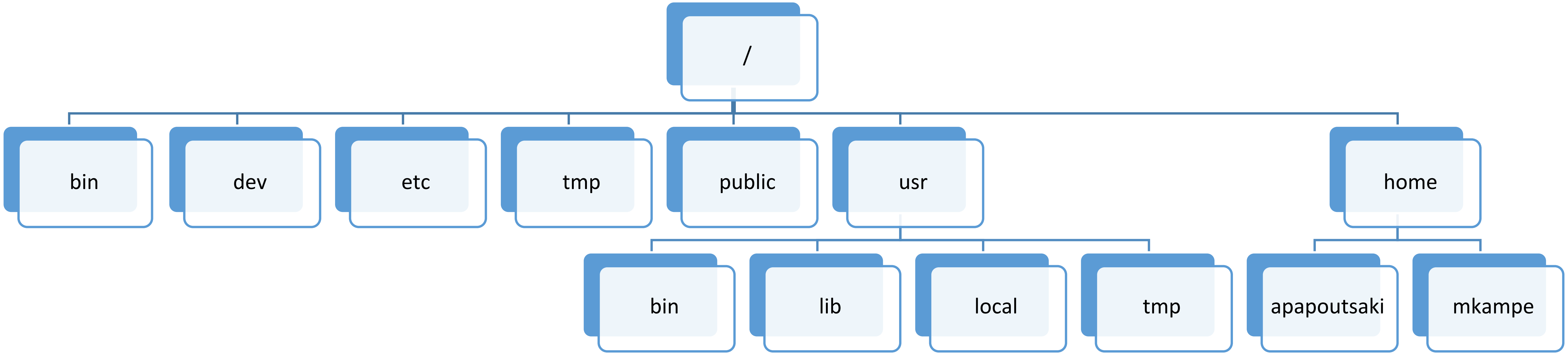
- Abstract data types that store elements hierarchically.
- Hierarchical: Each element in a tree has a parent (immediate ancestor) and zero or more children (immediate descendants).
- Appropriate when the linear, “before” and “after”, relationship is not enough.
- Trees in CS grow upside-down.



Definition of a tree

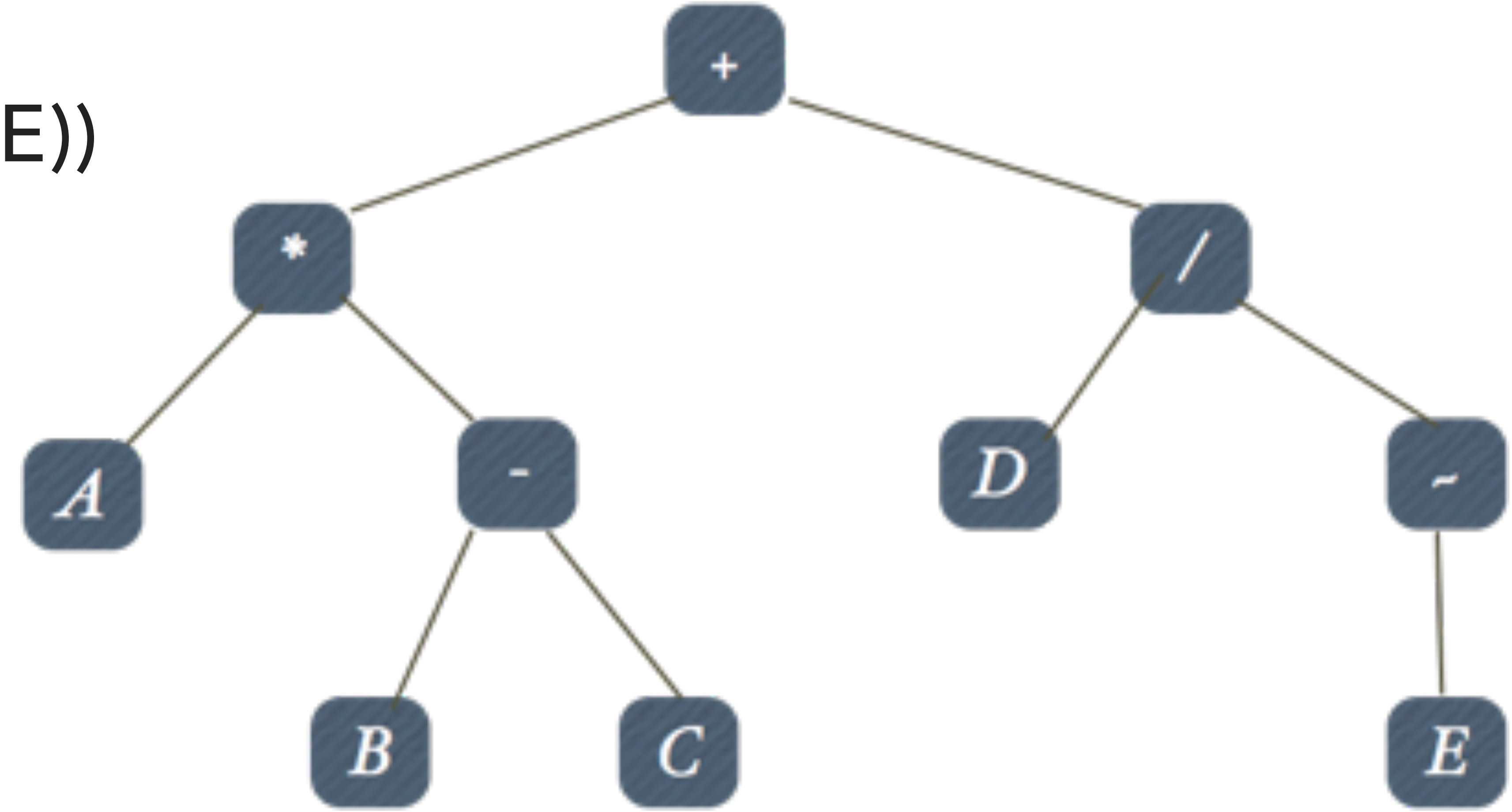
- A tree T is a set of nodes that store elements based on a parent-child relationship:
    - If T is non-empty, it has a node called the root of T, that has no parent.
    - Each node v, other than the root, has a unique parent node u. Every node with parent u is a child of u.

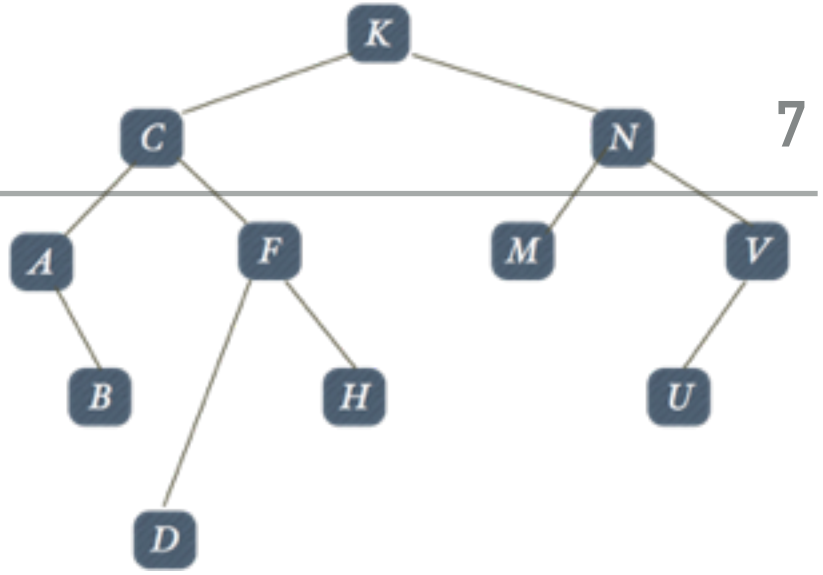
Example: Unix file system



Example: Expression tree

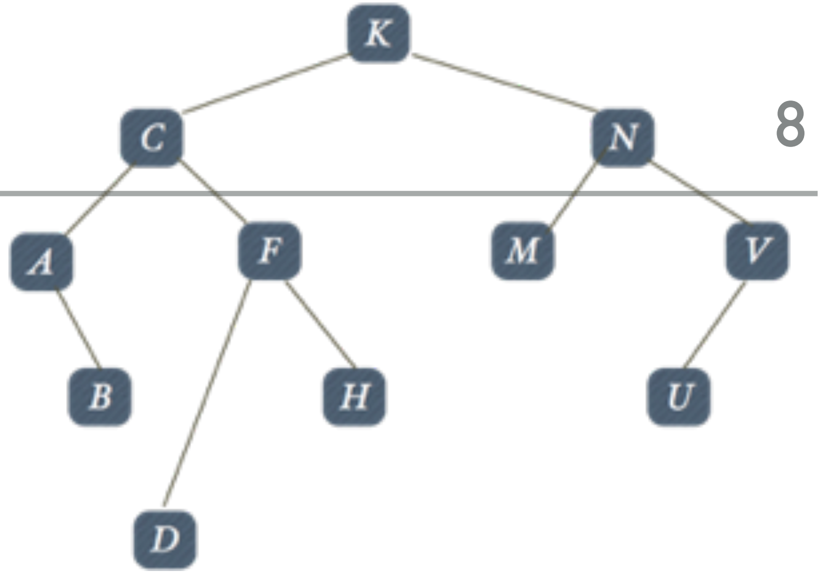
- If node is a leaf, then value is variable or constant.
- If node is internal, the value calculated by applying operations on its children.
- (A*(B-C))+(D/~E))





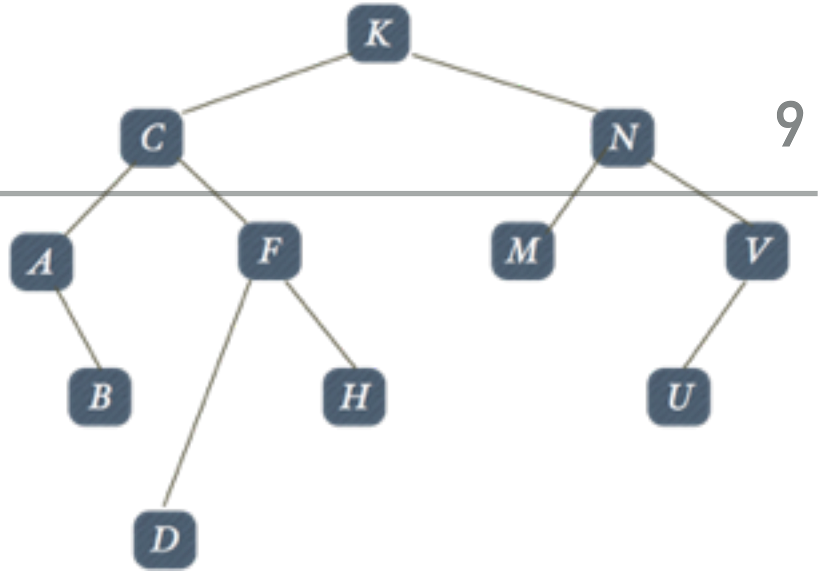
Family Tree Terminology

- Edge: a pair of nodes s.t. one is the parent of the other, e.g., (K,C).
- Parent node is directly above child node: K is parent of C and N.
- Sibling nodes have same parent, e.g., A and F.
- K is ancestor of B.
- B is descendant of K.
- Node plus all descendants gives subtree.
- Nodes without successors are called leaves or external. The rest are called internal.
- A set of trees is called a forest.



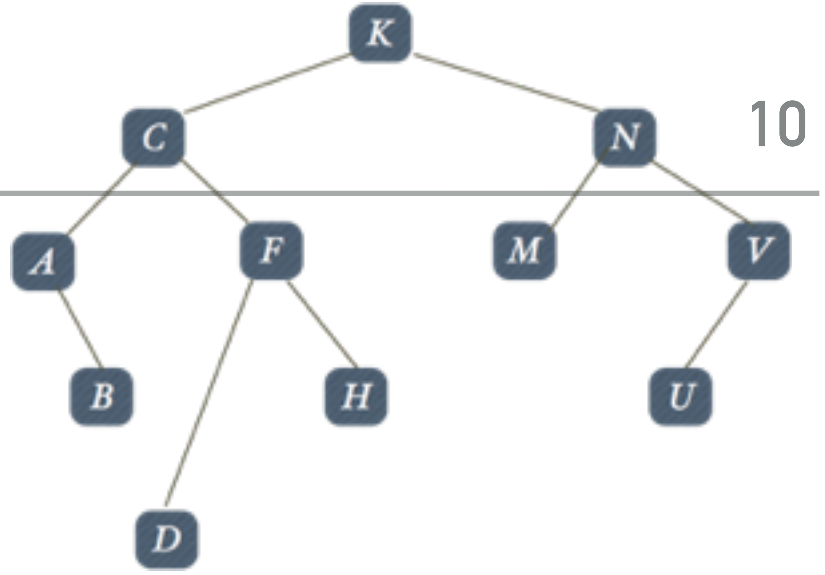
More Terminology

- Simple path: a series of distinct nodes s.t. there are edges between successive nodes.
- Path length: number of edges in path, e.g., path K-C-A has length 2.
- Height of node: length of longest path to leaf.
- Height of tree: height of root.
- Degree of node: number of its children.
- Degree of tree (arity): max degree of any of its nodes.
- Binary tree: a tree with arity of 2.



Even More Terminology

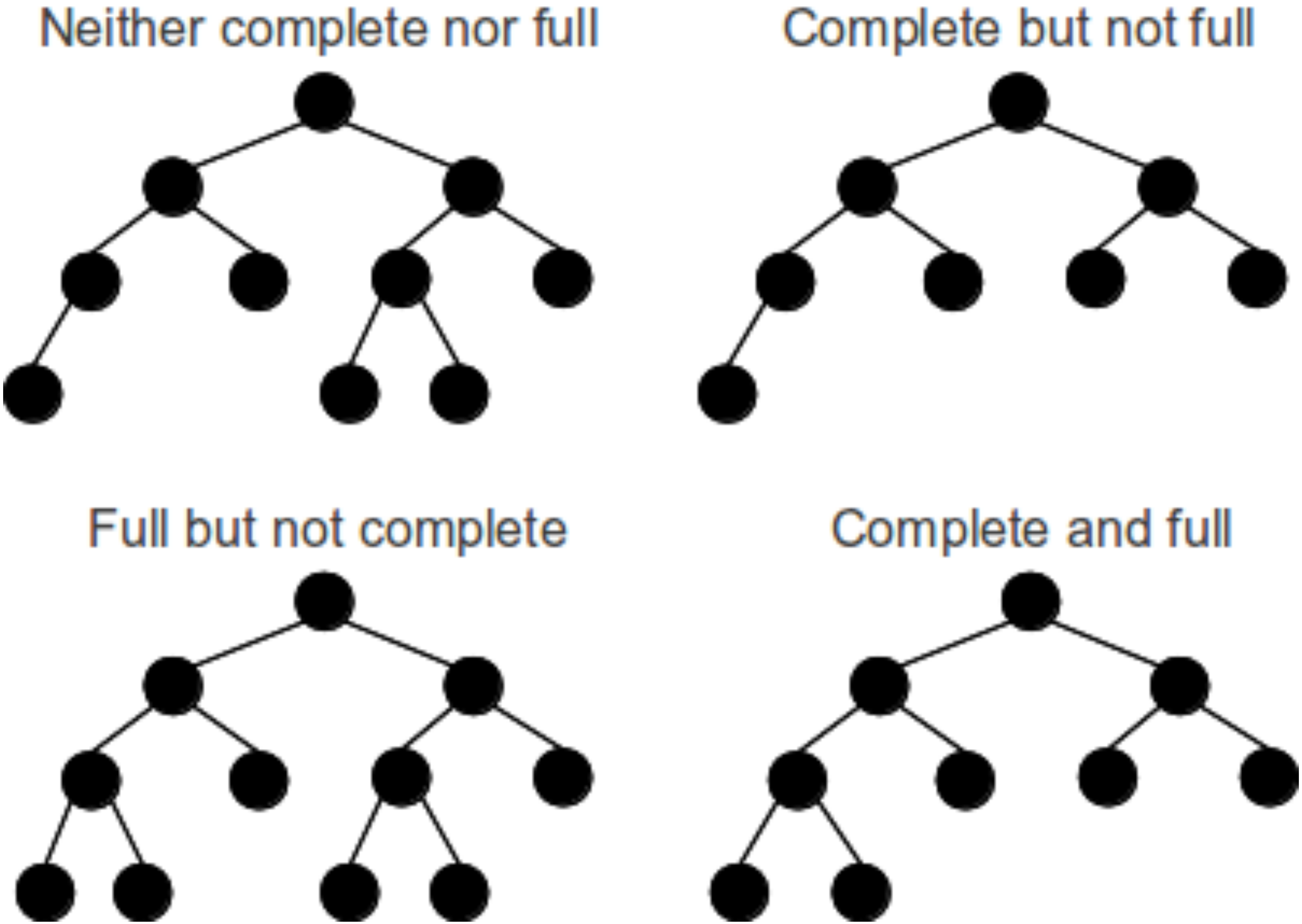
- Level/depth of node defined recursively:
  - Root is at level 0.
  - Level of any other node is equal to level of parent + 1.
  - It is also known as the length of path from root or number of ancestors.
- Height of node defined recursively:
  - If leaf, height is 0.
  - Else height is max height of child + 1.



But wait there's more!

- Full (or proper): a binary tree whose every node has 0 or 2 children.
- Complete: a binary tree with minimal height. Any holes in tree would appear at last level to right, i.e., all nodes of last level are as left as possible.

BINARY TREES



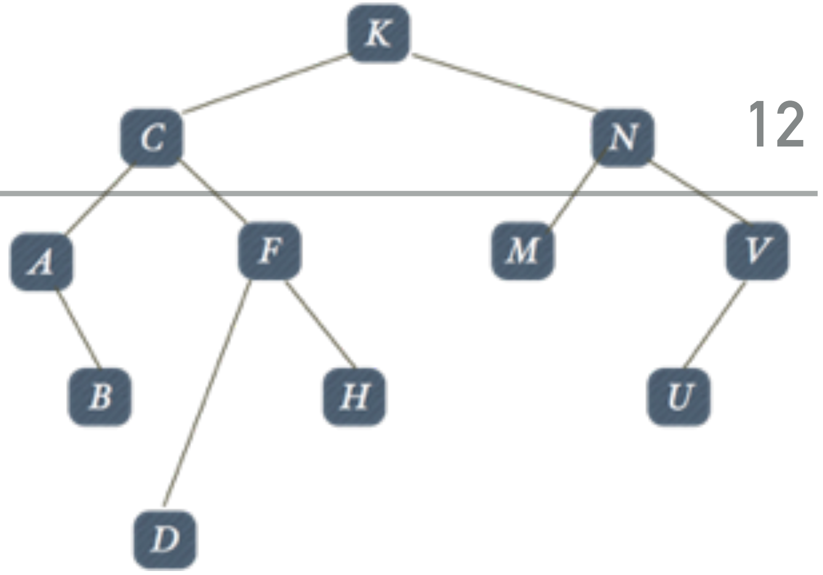
Neither complete nor full

Complete but not full

Full but not complete

Complete and full

http://code.cloudkaksha.org/binary-tree/types-binary-tree



Counting

- Lemma: if T is a binary tree, then at level k, T has $\leq 2^k$ nodes.
- Theorem: If T has height h, then # of nodes n in T satisfy: $h + 1 \leq n \leq 2^{h+1} - 1$.
- Equivalently, if T has n nodes, then $log(n+1) - 1 \leq h \leq n - 1$.

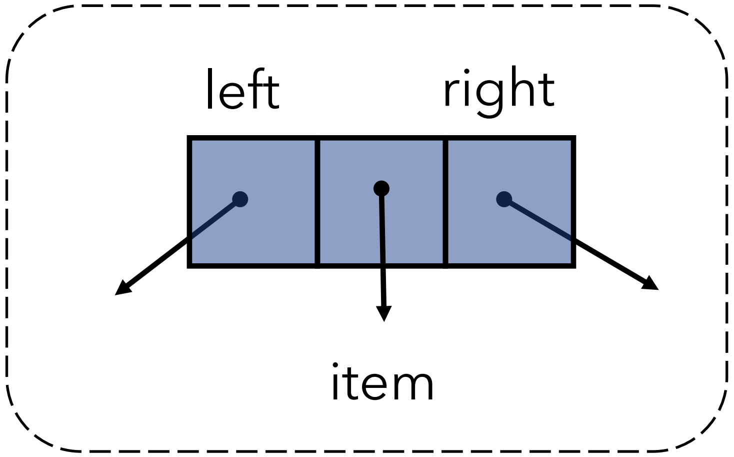
Basic idea behind a simple implementation

```
public class BinaryTree<Item> {
    private Node root;

    /**
     * A node subclass which contains various recursive methods
     *
     * @param <Item>  The type of the contents of nodes
     */
    private class Node {
        private Item item;

        private Node left;
        private Node right;

        /**
         * Node constructor with subtrees
         *
         * @param left   the left node child
         * @param right     the right node child
         * @param item   the item contained in the node
         */
        public Node(Node left, Node right, Item item) {
            this.left = left;
            this.right = right;
            this.item = item;
        }
```

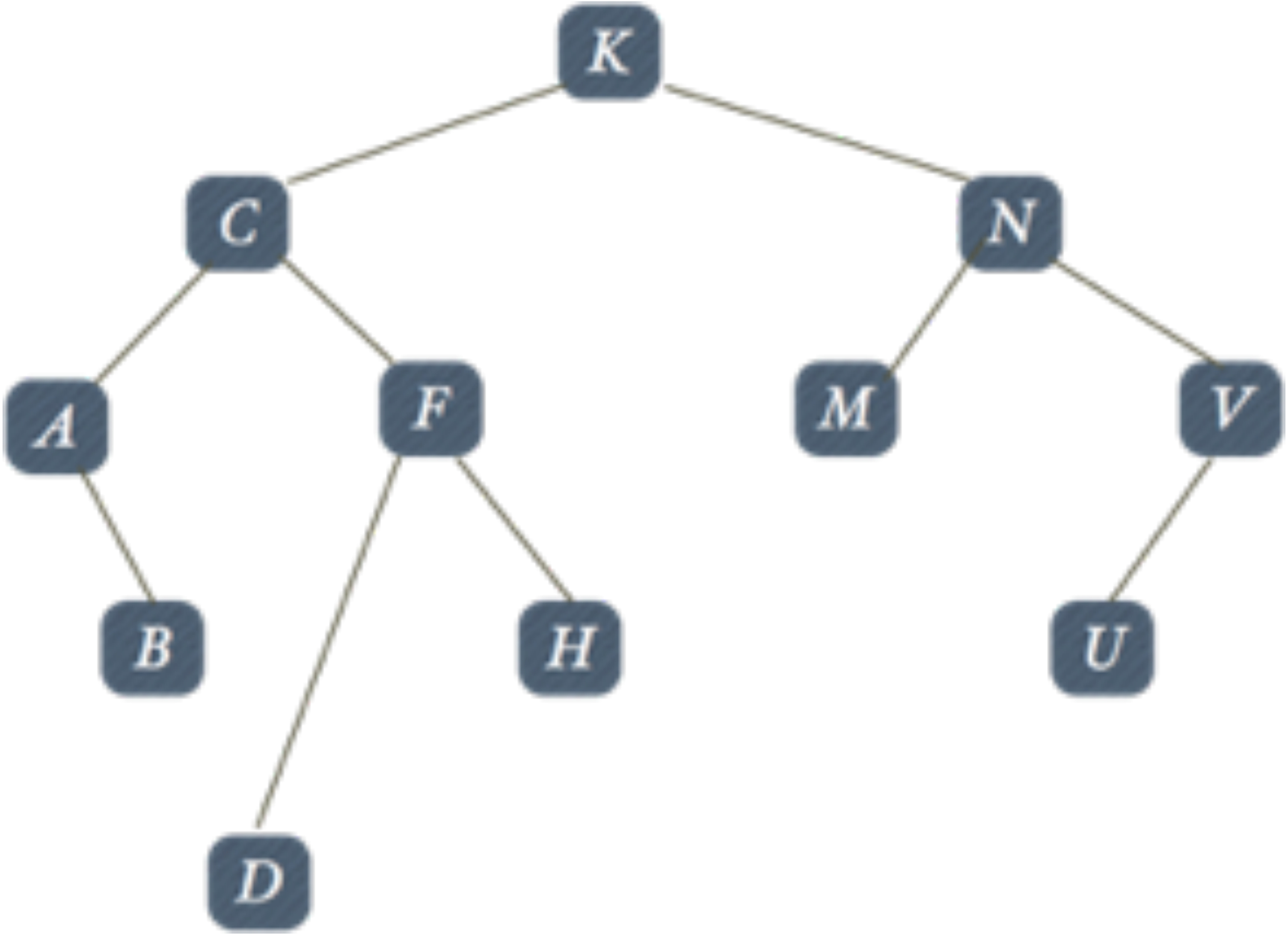


Lecture 22: Binary Trees

- Binary Trees
- Tree traversals

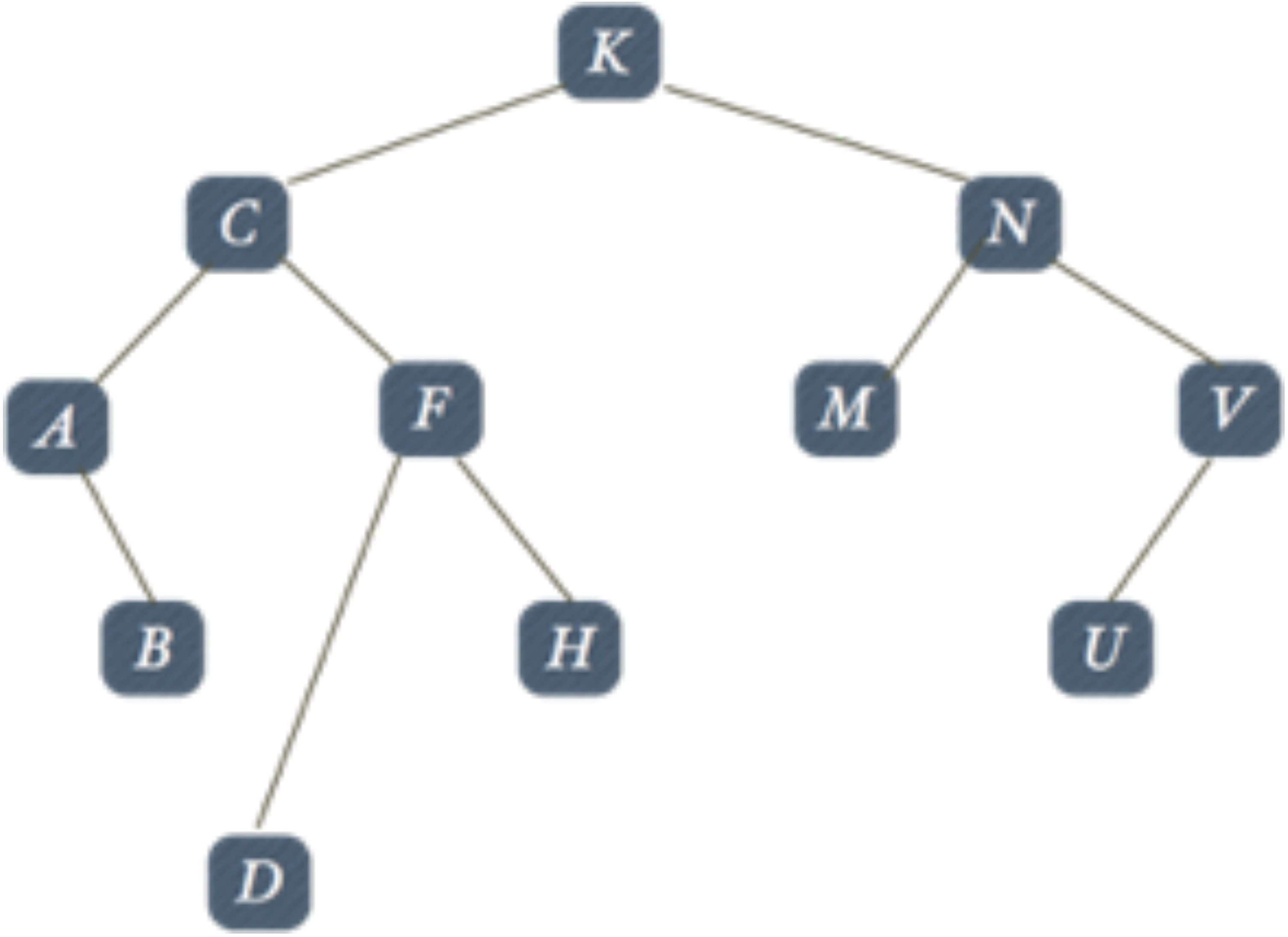
Pre-order traversal

- Root, Left Subtree, Right Subtree
- K C A B F D H N M V U



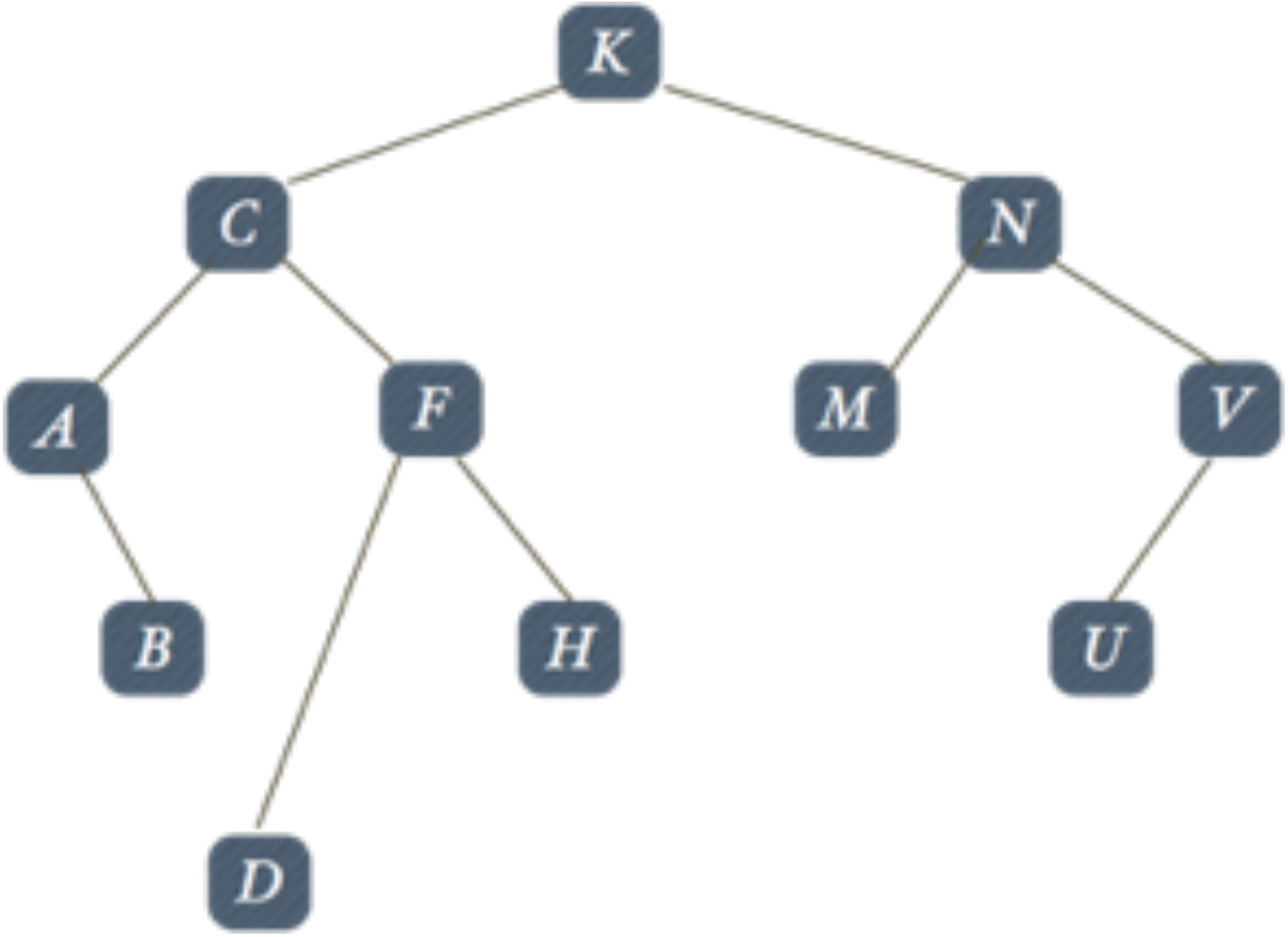
In-order traversal

- Left Subtree, Root, Right Subtree
- A B C D F H K M N U V



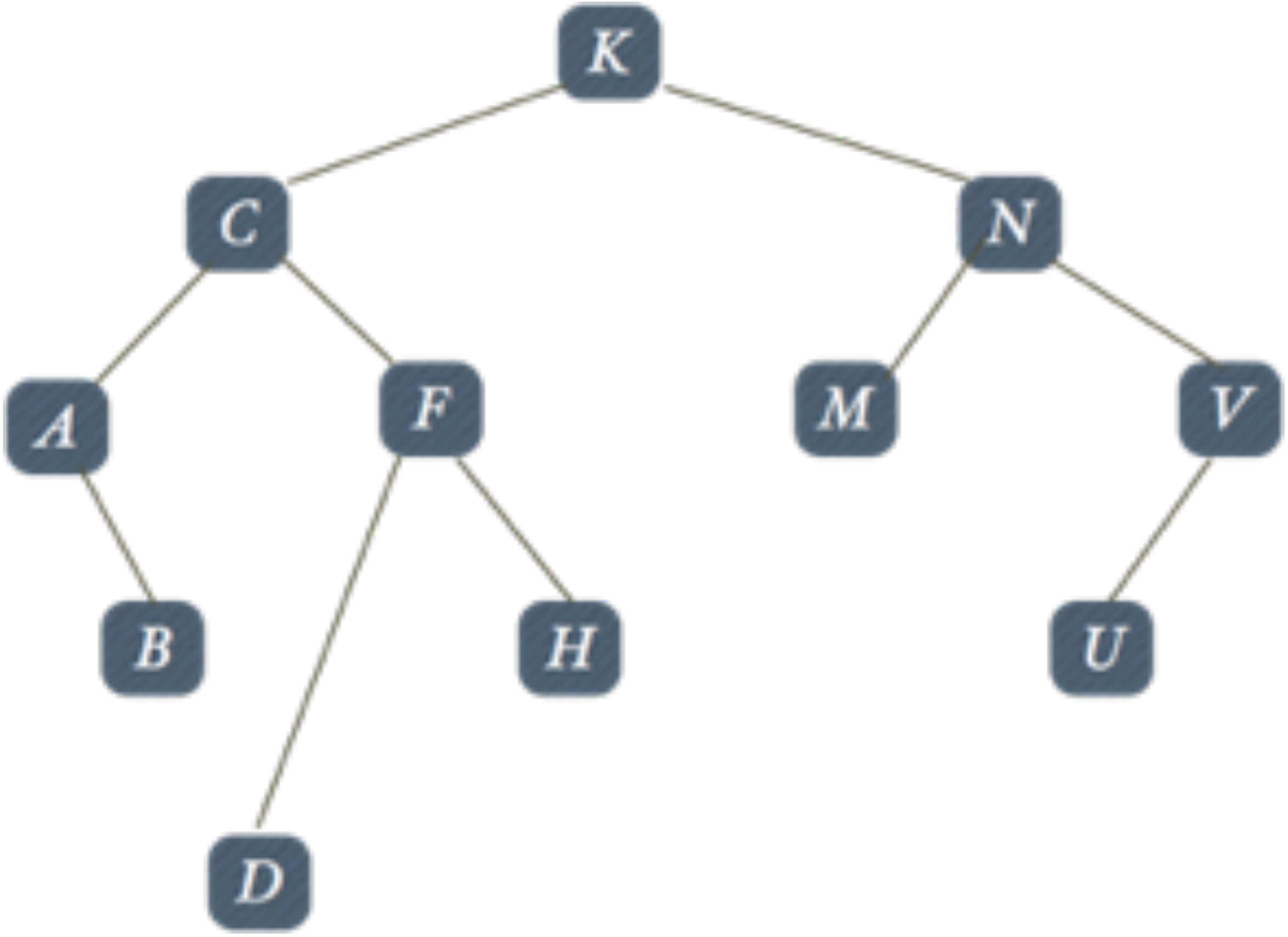
Post-order traversal

- Left Subtree, Right Subtree, Root
- B A D H F C M U V N K



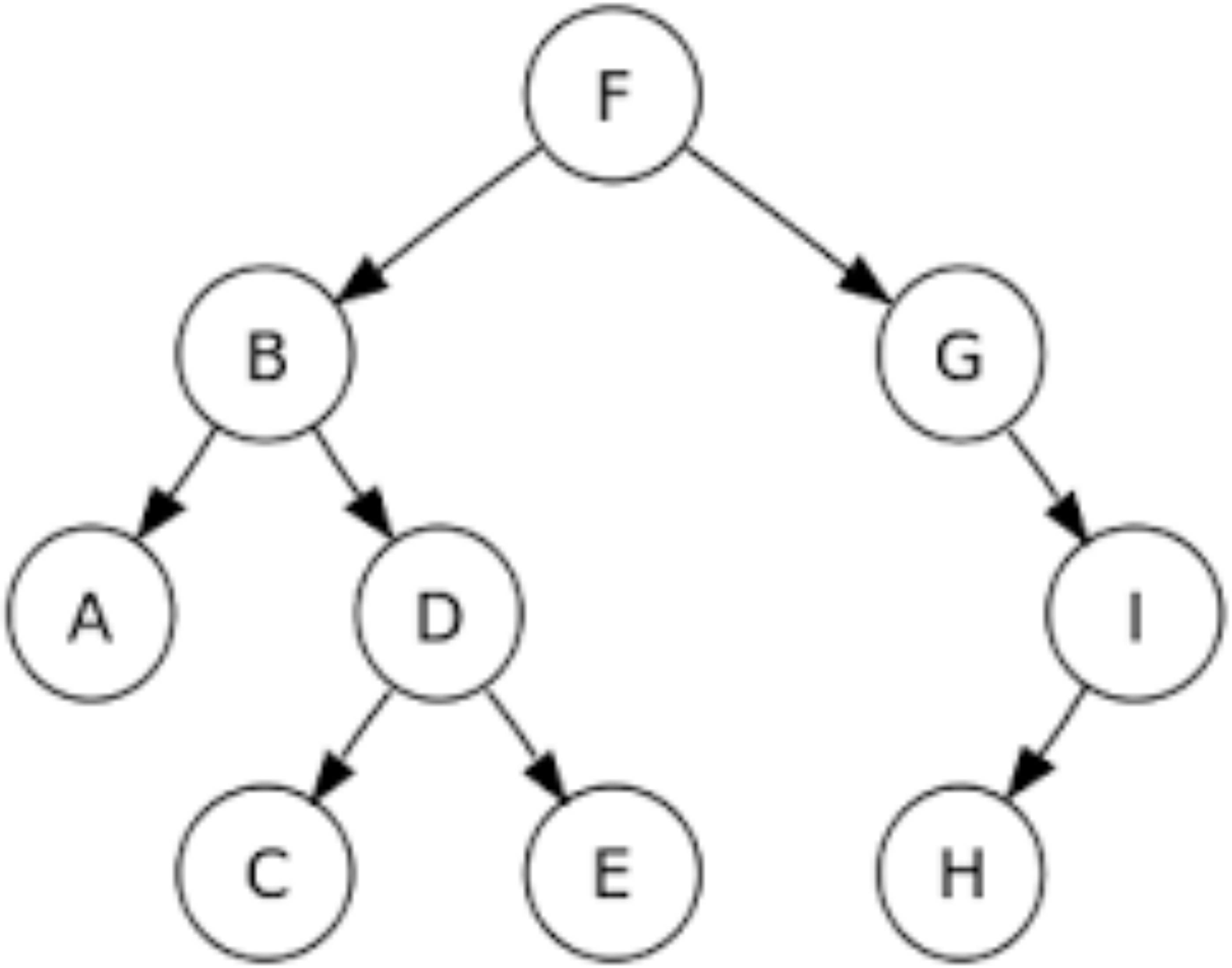
Level-order traversal

- All nodes of level i before nodes in level $i + 1$
- K C N A F M V B D H U



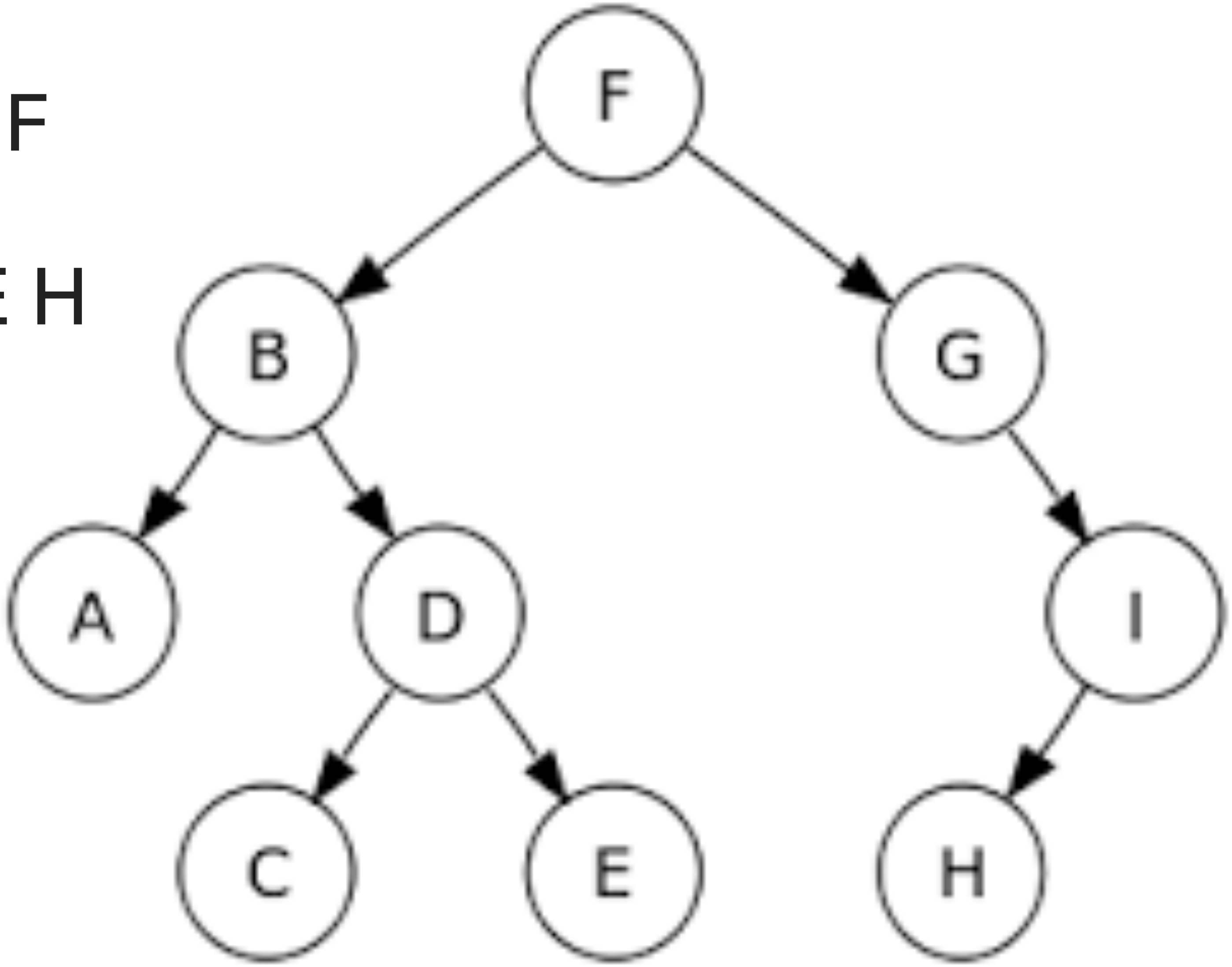
Practice Time

- List the nodes in pre-order, in-order, post-order, and level order.



Answer

- Pre-order: F B A D C E G I H
- In-order: A B C D E F G H I
- Post-order: A C E D B H I G F
- Level-order: F B G A D I C E H



Lecture 22: Binary Trees

- Binary Trees
- Tree traversals