

Binary Tree Traversal:

Breadth-First and Depth-First Strategies

Tree is not a linear data structure. There is no logical start with a pointer, such as a linked list.

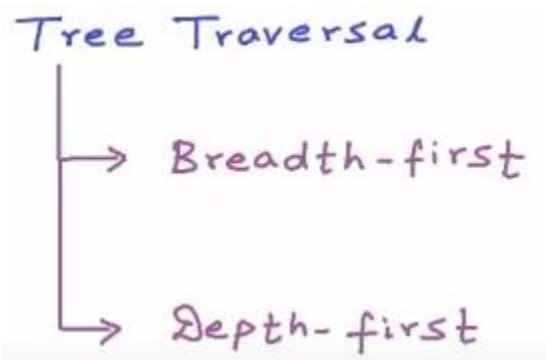
A Binary Tree can have multiple possible directions.

Tree Traversal

↳ process of visiting
each node in the tree
exactly once in some order.

... when we say “visiting a node,” we mean processing data in that node.

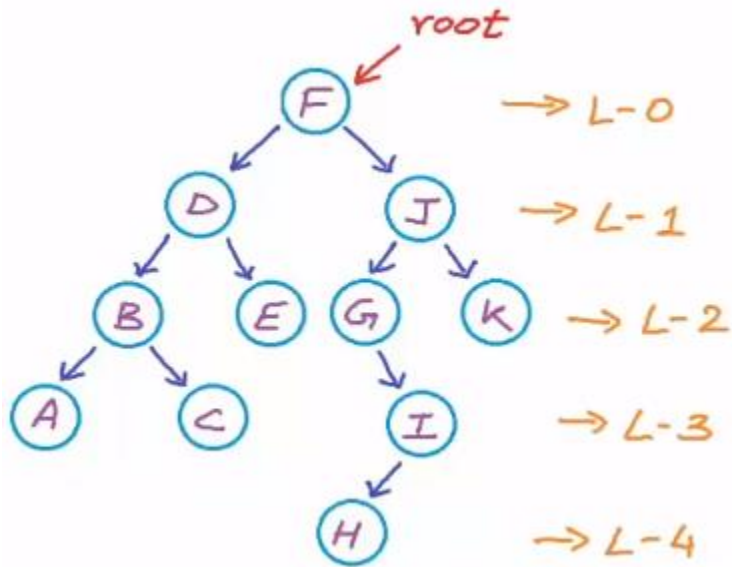
Tree Traversal can be classified into 2 categories:



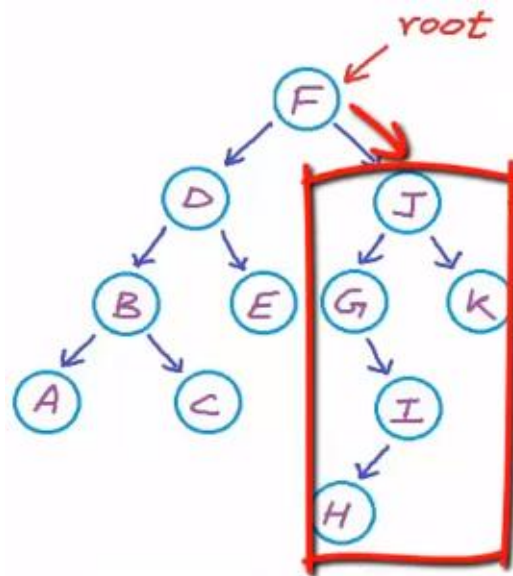
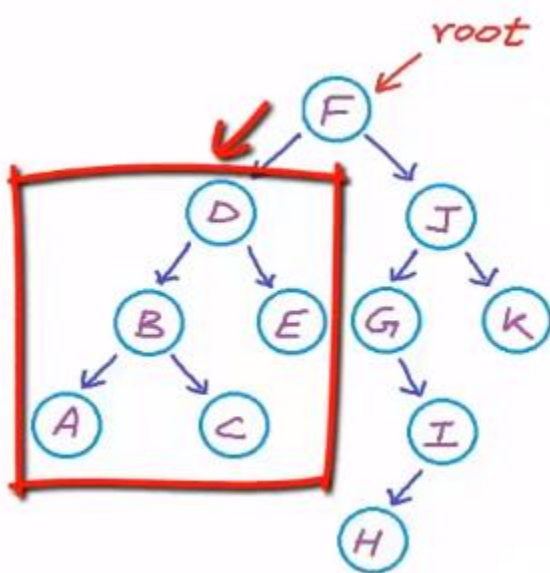
Breadth-First: We visit all nodes on the same level, before visiting nodes on the next level.

This is also known as **Level-Order Traversal**:

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



Depth-First: We visit a sub-tree (child and its respective grandchildren along one path) before visiting the other sub-tree:



There are 3 different types of Depth-First Strategies:

$\langle \text{root} \rangle \langle \text{left} \rangle \langle \text{right} \rangle$ - Preorder
 $\langle \text{left} \rangle \langle \text{root} \rangle \langle \text{right} \rangle$ - Inorder
 $\langle \text{left} \rangle \langle \text{right} \rangle \langle \text{root} \rangle$ - Postorder

Tree Traversal

- Breadth-first
 - ↳ Level-order
- Depth-first

<root> <left> <right> - Preorder

<left> <root> <right> - Inorder

<left> <right> <root> - Postorder