Day 14 Tree

ITSRUNTYM

1. What is a Tree?

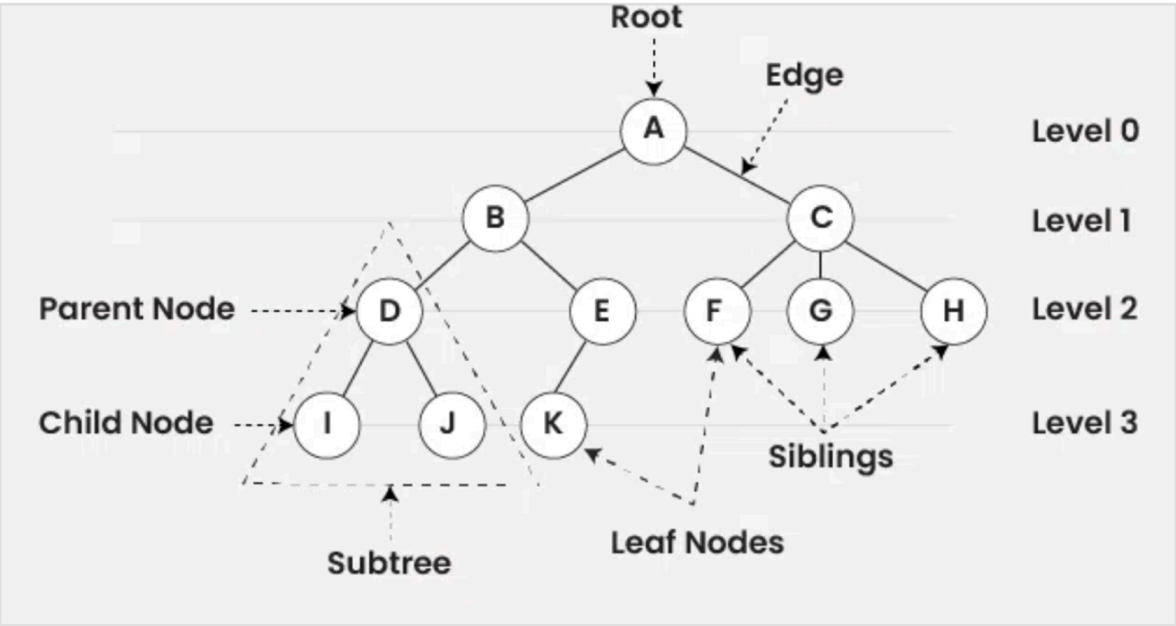
A **Tree** is a non-linear, hierarchical data structure made up of **nodes**. Each node has:

- **Data**
- References to child nodes

Basic Terminology:

Term	**Description**
Root	Topmost node of the tree
Child	Node descending from another
Parent	Node with children
Leaf	Node with no children
Sibling	Nodes with the same parent
Depth	Distance from the root
Height	Longest path to a leaf
Subtree	A tree within a tree

2. Tree Diagram



3. Types of Trees

Type	Description
Binary Tree	Each node has at most 2 children
Binary Search Tree (BST)	Left < Root < Right
Balanced Tree	Self-balancing BST
AVL Tree	Tree where height of subtrees differ by at most 1.
Red-Black Tree	Self-balancing BST with color rules
N-ary Tree	Node can have more than 2 children
Segment Tree	Used for range queries
Heap	Complete binary tree used for priority

4. Tree Traversals

Depth-First Search (DFS):

1. **Inorder (Left, Root, Right)**
2. **Preorder (Root, Left, Right)**
3. **Postorder (Left, Right, Root)**

Breadth-First Search (BFS):

- **Level Order** → Traverse level by level using a queue

#	Problem	Pattern	LeetCode Link
1	**Maximum Depth of Binary Tree**	DFS (Postorder)	🔗 Link
2	**Diameter of Binary Tree**	DFS + Return 2 Values	🔗 Link
3	**Same Tree**	Recursion	🔗 Link
4	**Symmetric Tree**	DFS or BFS	🔗 Link
5	**Invert Binary Tree**	Postorder DFS	🔗 Link
6	**Path Sum**	DFS + Target Tracking	🔗 Link
7	**Subtree of Another Tree**	Tree Traversal + Comparison	🔗 Link

8	**Lowest Common Ancestor of a Binary Tree**	Recursive DFS	🔗 Link
9	**Binary Tree Level Order Traversal**	BFS with Queue	🔗 Link
10	**Binary Tree Right Side View**	BFS + Right Priority	🔗 Link
11	**Construct Binary Tree from Preorder and Inorder Traversal**	Recursion + Divide & Conquer	🔗 Link
12	**Validate Binary Search Tree**	Inorder + Bounds Check	🔗 Link
13	**Kth Smallest Element in a BST**	Inorder Traversal	🔗 Link
14	**Convert Sorted Array to BST**	Divide and Conquer	🔗 Link
15	**Serialize and Deserialize Binary Tree**	Preorder or Level Order	🔗 Link