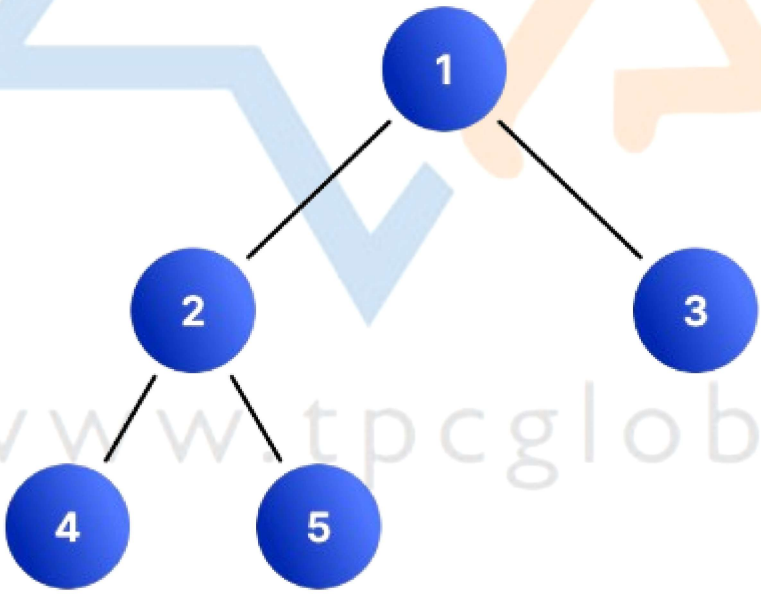
Binary Tree

Introduction

- A **Binary Tree** is a hierarchical data structure in which each node has at most **two children**:
    - **Left Child**
    - **Right Child**
- The **topmost node** is called the **Root.**
- A **Leaf Node** is a node that has no children.



Properties of Binary Tree

1. Max nodes at level L = 2L2^L2L.
2. Max nodes in a binary tree of height h = 2h+1−12^{h+1} - 12h+1−1.
3. For n nodes, min possible height = ⌊log2(n)⌋\lfloor \log_2(n) \rfloor⌊log2(n)⌋.
4. In a **binary tree**,
    - Number of leaf nodes = Number of nodes with 2 children + 1.

Types of Binary Trees

1. **Full Binary Tree** → Every node has either 0 or 2 children.
2. **Perfect Binary Tree** → All internal nodes have 2 children, and all leaves are at the same level.
3. **Complete Binary Tree** → All levels are filled except the last, which is filled from **left to right**.
4. **Skewed Binary Tree** → Tree where all nodes have only left children (left-skewed) or only right children (right-skewed).
5. **Balanced Binary Tree** → Difference between heights of left and right subtrees is at most 1.

Applications of Binary Trees

- **Expression Trees** (used in compilers).
- **Hierarchical representation** (file systems, XML/HTML parsing).
- **Binary Search Trees** → efficient searching, insertion, deletion.

- **Heaps/Priority Queues** → scheduling, shortest path algorithms.
- **Huffman Coding Tree** → data compression.