Trees – Binary Tree

1. Introduction

A **Binary Tree** is a special type of **tree data structure** in which **each node can have at most two children**.

These children are referred to as:

- **Left Child**
- **Right Child**

Binary trees form the foundation for many important data structures such as **Binary Search Trees, Heaps, and Expression Trees**.

---

2. What is a Binary Tree?

A binary tree is a hierarchical structure where:

- Each node contains data
- Each node has **0, 1, or 2 children**
- Children are categorized as left and right

There is **no restriction on the order of values** in a general binary tree.

---

3. Structure of a Binary Tree

```
      A
     / \
    B   C
   / \
  D   E
```

- A is the root
- B and C are children of A

- B has two children (D and E)
- C has no children

4. Basic Terminology (Binary Tree)

Term	Meaning
Root	Topmost node
Parent	Node having children
Child	Node connected below a parent
Leaf	Node with no children
Left Child	Node on the left side
Right Child	Node on the right side
Subtree	Tree formed from a node

5. Properties of a Binary Tree

- Maximum children per node = 2
- Each node has a left and/or right child
- No cycles
- Hierarchical structure
- Recursive nature (each subtree is also a binary tree)

6. Types of Binary Trees

1. Full Binary Tree

- Every node has **0 or 2 children**

2. Complete Binary Tree

- All levels are filled except possibly the last
- Nodes are filled from **left to right**

3. Perfect Binary Tree

- All internal nodes have 2 children
- All leaf nodes are at the same level

4. Skewed Binary Tree

- All nodes have only one child
- Can be left-skewed or right-skewed

7. Binary Tree vs Binary Search Tree

Feature	Binary Tree	Binary Search Tree
Node order	No order	Left < Root < Right
Searching	Not efficient	Efficient
Structure	General	Ordered

8. Representation of Binary Tree

Binary trees can be represented using:

- **Linked representation** (nodes with left and right pointers)
- **Array representation** (mainly for complete binary trees)

9. Binary Tree Traversals (Overview)

Traversal means visiting all nodes.

Common traversals:

- **Preorder** (Root → Left → Right)
- **Inorder** (Left → Root → Right)
- **Postorder** (Left → Right → Root)
- **Level Order** (Breadth First)

(Each traversal serves a different purpose.)

10. Advantages of Binary Tree

- Represents hierarchical data
- Efficient insertion and deletion
- Basis for many advanced data structures
- Flexible structure

11. Limitations of Binary Tree

- No guaranteed search efficiency
- Can become skewed
- Requires careful traversal logic
- Uses more memory due to pointers

12. Applications of Binary Tree

- Expression evaluation
- Decision-making systems
- File directory structure
- Syntax trees
- Artificial Intelligence

13. Real-World Example

A **family tree**:

- Each person can have at most two parents
- Hierarchical relationship is maintained

14. Summary

- Binary tree allows at most two children per node

- Children are left and right
- No ordering rule in general binary tree
- Forms base for advanced trees
- Widely used in computer science