## **4. Types of Binary Tree**

### **4.1 Full Binary Tree**

**Definition:** Every node has **0 or 2 children**. No node has only one child.

**Example Diagram:**

1

/ \

2 3

/ \

4 5

**Properties:**

Nodes with children always have two children.

Not all levels need to be completely filled.

### **4.2 Complete Binary Tree**

**Definition:** All levels are **completely filled except possibly the last**, and all nodes in the last level are **as left as possible**.

**Example Diagram:**

1

/ \

2 3

/ \ /

4 5 6

**Properties:**

Efficient for array representation.

Often used in **heap structures**.

### **4.3 Perfect Binary Tree**

**Definition:** A binary tree that is **both full and complete**.

All internal nodes have **two children**, and all leaf nodes are at the **same level**.

**Example Diagram:**

1

/ \

2 3

/ \ / \

4 5 6 7

**Properties:**

Number of nodes = 2^(h+1) - 1 (h = height of tree).

### **4.4 Balanced Binary Tree**

**Definition:** A binary tree in which **the height difference between left and right subtrees of every node is ≤ 1**.

**Example Diagram:**

10

/ \

5 15

/ \ \

2 7 20

**Properties:**

Ensures efficient operations (O(log n) height).

Used in **AVL trees** and **Red-Black trees**.

### **4.5 Skewed Binary Tree**

**Definition:** A binary tree in which **all nodes have only one child**.

Can be **Left Skewed** or **Right Skewed**.

**Left Skewed Example:**

10

/

9

/

8

**Right Skewed Example:**

10

\

11

\

12

**Properties:**

Height = number of nodes (worst-case).

Poor performance for operations like search in BST.