**Binary Trees:**

A binary tree is a hierarchical data structure composed of nodes, where each node has at most two children: a left child and a right child. The topmost node in a binary tree is called the root node. Each node in the tree can have zero, one, or two children. If a node has no children, it is called a leaf node.

Binary Search Trees (BST):

A binary search tree (BST) is a specific type of binary tree in which the values of all nodes in the left subtree are less than the value of the node, and the values of all nodes in the right subtree are greater than the value of the node. This property makes searching, insertion, and deletion operations efficient in a BST.

Balanced Trees:

Balanced trees are binary trees in which the heights of the left and right subtrees of any node differ by at most one. These trees ensure that operations such as insertion, deletion, and search have a worst-case time complexity of O(log n), where n is the number of nodes in the tree. Examples of balanced trees include AVL trees and Red-Black trees.

* AVL Trees: AVL trees are self-balancing binary search trees in which the heights of the left and right subtrees of every node differ by at most one. AVL trees maintain balance through rotations during insertion and deletion operations.
* Red-Black Trees: Red-Black trees are another type of self-balancing binary search tree. Each node in a Red-Black tree is assigned a color (either red or black), and the tree maintains certain properties to ensure balance. Red-Black trees use color changes and rotations to maintain balance.

Tree Traversal Algorithms:

Traversal algorithms are used to visit and process all nodes in a tree in a specific order. There are three common traversal algorithms for binary trees:

* In-order Traversal: Visit the left subtree, then the root node, and finally the right subtree.
* Pre-order Traversal: Visit the root node, then the left subtree, and finally the right subtree.
* Post-order Traversal: Visit the left subtree, then the right subtree, and finally the root node.

These traversal algorithms are fundamental for performing various operations on binary trees, such as printing the tree elements, evaluating expressions, and constructing an expression tree.

Binary Heap:

A binary heap is a complete binary tree where every parent node has a value less than or equal to (for a min heap) or greater than or equal to (for a max heap) the values of its children. Binary heaps are commonly used to implement priority queues, where elements with higher priority are served before elements with lower priority.

* Min Heap: In a min heap, the value of each parent node is less than or equal to the values of its children.
* Max Heap: In a max heap, the value of each parent node is greater than or equal to the values of its children.