1. **Preorder Traversal**
   * **Order:** Visit the root node → Traverse the left subtree → Traverse the right subtree.
   * **Output Sequence:** Root → Left → Right.
   * **Use Cases:**
     + Creating a copy of the tree.
     + Prefix expression evaluation (e.g., used in compilers and expression trees).
   * **Example:** For the tree

1

/ \

2 3

/ \

4 5

Preorder traversal: 1, 2, 4, 5, 3.

1. **Inorder Traversal**
   * **Order:** Traverse the left subtree → Visit the root node → Traverse the right subtree.
   * **Output Sequence:** Left → Root → Right.
   * **Use Cases:**
     + Sorting nodes in ascending order (for binary search trees).
     + Used in expression trees to produce infix notation.
   * **Example:** For the tree:

1

/ \

2 3

/ \

4 5

Inorder traversal: 4, 2, 5, 1, 3.

1. **Postorder Traversal**
   * **Order:** Traverse the left subtree → Traverse the right subtree → Visit the root node.
   * **Output Sequence:** Left → Right → Root.
   * **Use Cases:**
     + Deleting or freeing nodes (e.g., cleaning up resources).
     + Postfix expression evaluation.
   * **Example:** For the tree:

1

/ \

2 3

/ \

4 5

Postorder traversal: 4, 5, 2, 3, 1.

In Sorted order \* if it's sorted by default \*