Preorder, Inorder, and Postorder

**Preorder Traversal (Root -> Left -> Right)** Preorder traversal means we visit nodes in this order:

1. Start at the root node.
2. Visit the left subtree.
3. Visit the right subtree.

**Simple Algorithm:**

Preorder (Node root):

if root is not null:

print(root.value)

Preorder(root.left)

Preorder(root.right)

**Example:** For this tree:

A

/ \

B C

/ \ \

D E F

Preorder traversal will be: **A B D E C F**

**Inorder Traversal (Left -> Root -> Right)** In inorder traversal, we visit the nodes in this order:

1. Visit the left subtree.
2. Visit the root node.
3. Visit the right subtree.

**Simple Algorithm:**

Inorder (Node root):

if root is not null:

Inorder(root.left)

print(root.value)

Inorder(root.right)

**Example:** For the same tree: Inorder traversal will be: **D B E A C F**

**Postorder Traversal (Left -> Right -> Root)** In postorder traversal, the order is:

1. Visit the left subtree.
2. Visit the right subtree.
3. Visit the root node.

**Simple Algorithm:**

Postorder (Node root):

if root is not null:

Postorder(root.left)

Postorder(root.right)

print(root.value)

**Example:** For the same tree: Postorder traversal will be: **D E B F C A**