**Definition of Tree Traversal**

Tree traversal is the method of visiting each node in a tree data structure one time. It allows us to access or process the information stored in each node. Different traversal techniques, such as depth-first and breadth-first, determine the order in which nodes are visited.

**A diagram of a tree

Description automatically generatedTree Traversal Techniques:**

**Inorder Traversal**

* **A diagram of a diagram

  Description automatically generated**Inorder traversal visits the node in the order: **Left -> Root -> Right**

**Algorithm for Inorder Traversal:**

* *Traverse the left subtree*
* *Visit the root.*
* *Traverse the right subtree*

**Time Complexity:**O(N)

**PreOrder Traversal**

* **A diagram of a diagram

  Description automatically generated**Preorder traversal visits the node in the order: **Root -> Left -> Right**

**Algorithm for Preorder Traversal:**

* *Visit the root.*
* *Traverse the left subtree,*
* *Traverse the right subtree*

**Time Complexity:**O(N)

**Postorder Traversal**

* Postorder traversal visits the node in the order: **Left -> Right -> Root**

**A diagram of a network

Description automatically generated**

**Algorithm for Postorder Traversal:**

* *Traverse the left subtree*
* *Traverse the right subtree*
* *Visit the root*

### **When should I use each traversal technique?**

1. Inorder traversal is often used for retrieving sorted data from a binary search tree (BST)

* Imagine a phone book organized as a binary search tree with contacts like Alice, Bob, Charlie, David, and Eve. Using inorder traversal, you retrieve the names in alphabetical order: Alice, Bob, Charlie, David, and Eve, along with their phone numbers. This makes it easy to find and look up any contact quickly!

1. Preorder traversal is useful for creating a copy of the tree.

* Imagine you're copying a family tree. You first write down the parent (like "Mom"), then you move to each child (like "Alice" and "Bob"). By visiting the parent first, you ensure the family structure is preserved as you replicate it.

1. Evaluating expressions in expression trees.

* **Example:** Think of a calculator that uses an expression tree to evaluate the expression ((3 + 4) \* 5). Postorder traversal would allow you to compute the result by first evaluating the left subtree (3 + 4), then the right subtree (5), and finally combining them (7 \* 5 = 35).