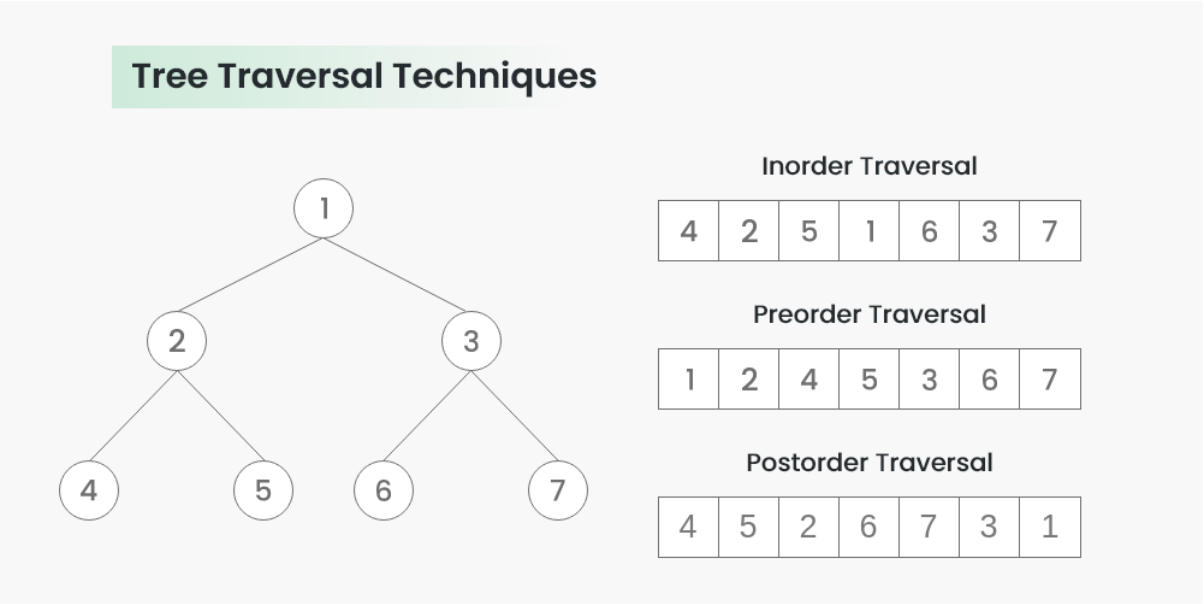
Traversal data structure

Traversal is a common operation performed on data structures. It is the process in which each and every element present in a data structure is visited or accessed) at least once.

A Tree Data Structure can be traversed in following ways:

1. *Depth First Search or DFS*
    1. *Inorder Traversal*
    2. *Preorder Traversal*
    3. *Postorder Traversal*
2. Level Order Traversal or Breadth First Search or BFS
3. Boundary Traversal
4. Diagonal Traversal

Depth First Search or DFS



Inorder Traversal

Algorithm Inorder(tree)

1. *Traverse the left subtree, i.e., call Inorder(left->subtree)*
2. *Visit the root.*
3. *Traverse the right subtree, i.e., call Inorder(right->subtree)*

Uses of Inorder Traversal:
In the case of binary search trees (BST), Inorder traversal gives nodes in non-decreasing order. To get nodes of BST in non-increasing order, a variation of Inorder traversal where Inorder traversal is reversed can be used.

Time Complexity: O(N)
Auxiliary Space: If we don't consider the size of the stack for function calls then O(1) otherwise O(h) where h is the height of the tree.

Preorder Traversal

Algorithm Preorder(tree)

1. *Visit the root.*
2. *Traverse the left subtree, i.e., call Preorder(left->subtree)*
3. *Traverse the right subtree, i.e., call Preorder(right->subtree)*

Uses of Preorder:
Preorder traversal is used to create a copy of the tree. Preorder traversal is also used to get prefix expressions on an expression tree.

Time Complexity: O(N)
Auxiliary Space: If we don't consider the size of the stack for function calls then O(1) otherwise O(h) where h is the height of the tree.

Postorder Traversal

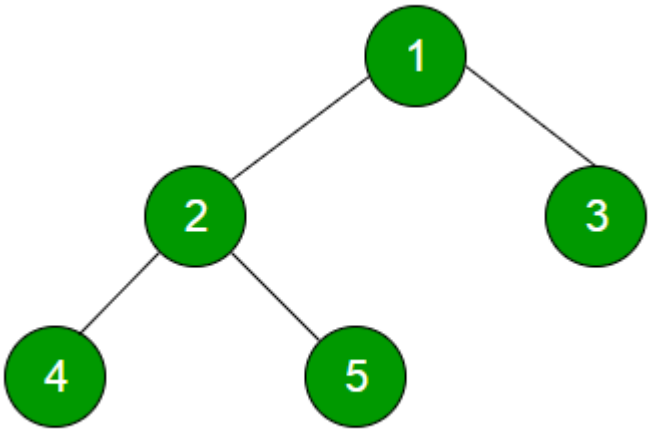
Algorithm Postorder(tree)

1. *Traverse the left subtree, i.e., call Postorder(left->subtree)*
2. *Traverse the right subtree, i.e., call Postorder(right->subtree)*
3. *Visit the root*

Uses of Postorder:
Postorder traversal is used to delete the tree. Please see the question for the deletion of a tree for details. Postorder traversal is also useful to get the postfix expression of an expression tree

Level Order Treversal *or Breadth First Search*

For each node, first, the node is visited and then it's child nodes are put in a FIFO queue. Then again the first node is popped out and then it's child nodes are put in a FIFO queue and repeat until queue becomes empty



Level Order Treversal:

1

2 3

4 5

Time Complexity: O(N)

The time complexity of BFS is O(V + E), where V is the number of nodes and E is the number of edges.