

Tree Traversal Techniques - GeeksforGeeks

Source: <https://www.geeksforgeeks.org/tree-traversals-inorder-preorder-and-postorder/>

Courses Tutorials Practice Jobs DSA Tutorial Interview Questions Quizzes Must Do Advanced DSA System Design Aptitude Puzzles Interview Corner DSA Python Technical Scripter 2026 Explore DSA Fundamentals Logic Building Problems Analysis of Algorithms Data Structures Array Data Structure String in Data Structure Hashing in Data Structure Linked List Data Structure Stack Data Structure Queue Data Structure Tree Data Structure Graph Data Structure Trie Data Structure Algorithms Searching Algorithms Sorting Algorithms Introduction to Recursion Greedy Algorithms Tutorial Graph Algorithms Dynamic Programming or DP Bitwise Algorithms Advanced Segment Tree Binary Indexed Tree or Fenwick Tree Square Root (Sqrt) Decomposition Algorithm Binary Lifting Geometry Interview Preparation Interview Corner GfG160 Practice Problem GeeksforGeeks Practice - Leading Online Coding Platform Problem of The Day - Develop the Habit of Coding DSA Course 90% Refund Tree Traversal Techniques Last Updated : 6 Dec, 2025 Tree traversal refers to the process of visiting or accessing each node of a tree exactly once in a specific order. Unlike linear data structures such as arrays, linked lists, or queues (which have only one logical way of traversal), trees offer multiple ways to traverse their nodes. Tree traversals are broadly classified into two categories: Depth-First Traversal (DFT) Explores as far as possible along a branch before exploring the next branch. Types: Inorder, Preorder, Postorder Breadth-First Traversal (BFT) Explores nodes level by level from top to bottom. Type: Level Order Traversal. The level order traversal of the above tree is 1, 2, 3, 4, 5, 6, and 7. Inorder Traversal Inorder traversal visits the node in the order: Left -> Root -> Right Algorithm for Inorder Traversal Inorder(tree) ● Traverse the left subtree, i.e., call Inorder(left->subtree) ● Visit the root. ● 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. Inorder traversal can be used to evaluate arithmetic expressions stored in expression trees. Also Check: Refer Inorder Traversal of Binary Tree for more Preorder Traversal Preorder traversal visits the node in the order: Root -> Left -> Right Algorithm for Preorder Traversal Preorder(tree) ● Visit the root. ● Traverse the left subtree, i.e., call Preorder(left->subtree) ● Traverse the right subtree, i.e., call Preorder(right->subtree) Uses of Preorder Traversal Preorder traversal is used to create a copy of the tree. Preorder traversal is also used to get prefix expressions on an expression tree. Also Check: Refer Preorder Traversal of Binary Tree for more Postorder Traversal Postorder traversal visits the node in the order: Left -> Right -> Root Algorithm for Postorder Traversal: Postorder(tree) ● Traverse the left subtree, i.e., call Postorder(left->subtree) ● Traverse the right subtree, i.e., call Postorder(right->subtree) ● Visit the root Uses of Postorder Traversal Postorder traversal is used to delete the tree. Postorder traversal is also useful to get the postfix expression of an expression tree. Postorder traversal can help in garbage collection algorithms, particularly in systems where manual memory management is used. Also Check : Refer Postorder Traversal of Binary Tree for more Level Order Traversal Level Order Traversal visits all nodes present in the same level completely before visiting the next level. Algorithm for Level Order Traversal LevelOrder(tree) ● Create an empty queue Q ● Enqueue the root node of the tree to Q ● Loop while Q is not empty ■ Dequeue a node from Q and visit it ■ Enqueue the left child of the dequeued node if it exists ■ Enqueue the right child of the dequeued node if it exists . Uses of Level Traversal Level-wise node processing, like finding maximum/minimum at each level. Tree serialization/deserialization for efficient storage and reconstruction. Solving problems like calculating the "maximum width of a tree" by processing nodes level by level. Also Check : Refer Level Order Traversal (Breadth First Search or BFS) of Binary Tree for more Other Tree Traversals Boundary Traversal Diagonal Traversal 1. Boundary Traversal Boundary Traversal of a Tree includes left boundary (nodes on left excluding leaf nodes) leaves (consist of only the leaf nodes) right boundary (nodes on right excluding leaf nodes) Also Check: Refer Boundary Traversal of binary tree for more 2. Diagonal Traversal In the Diagonal Traversal of a Tree, all the nodes in a single diagonal will be printed one by one. Also Check: Refer Diagonal Traversal of Binary Tree for more Related Articles Construct Tree from given Inorder and Preorder traversals DSA Tutorial Tree Coding Problems for Interviews Comment Article Tags: Article Tags: Tree DSA Trees Inorder Traversal Preorder Traversal PostOrder Traversal Snapdeal tree-traversal Tutorials + 5 More

