

Tree Traversal using Level Order Traversal

Mr. Vikas Kumar

Assistant Professor

Industry Embedded Program

Lesson Plan

Subject/Course	Competitive Coding
Lesson Title	Tree Traversal using Level Order Traversal

Lesson Objectives

Understand the concept of Level Order Traversal in a binary tree.

Learn how to process tree nodes level by level using a queue.

Implement Level Order Traversal using Breadth-First Search (BFS) logic.

Analyze the time and space complexity of the algorithm for different tree structures.

Problem Statement:

Level Order Traversal of a binary tree means visiting all nodes level by level, from top to bottom and left to right at each level.

The goal is to print the elements in this order.

Concept

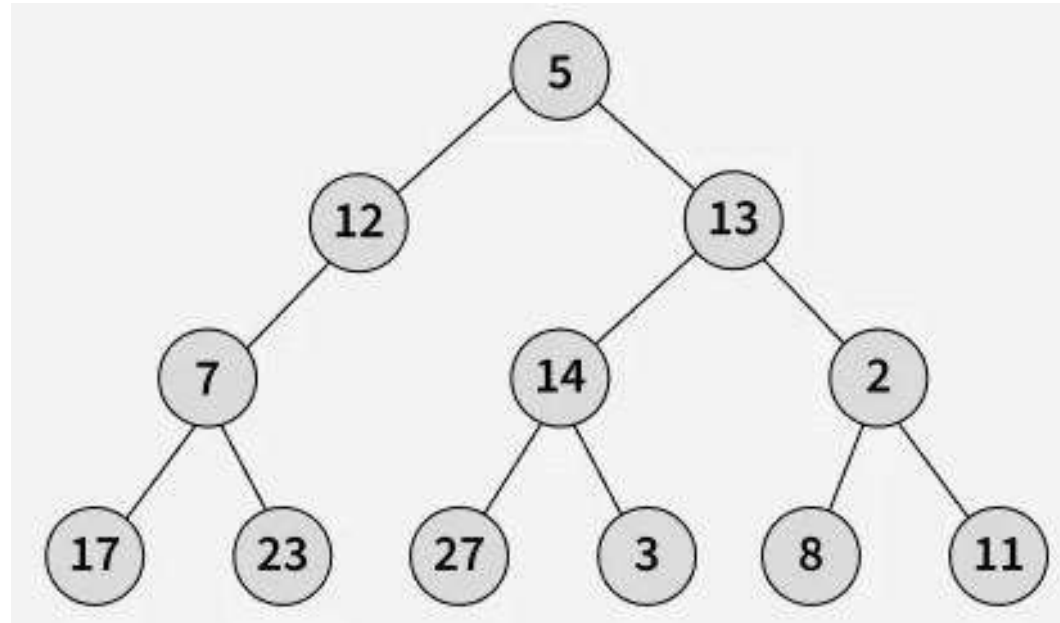
- Level Order Traversal is based on Breadth-First Search (BFS).
- We use a queue to store nodes of each level.
- Nodes are visited in the order they appear in the queue.
- Commonly used in shortest path problems and hierarchical data processing.

Algorithm/Logic

Steps:

1. Create an empty queue and enqueue the root node.
2. While the queue is not empty:
 - Dequeue a node, print its data.
 - Enqueue its left and right child (if present).
3. Continue until all nodes are processed.

Visualization



Output: [[5], [12, 13], [7, 14, 2], [17, 23, 27, 3, 8, 11]]

Explanation: Start with the root - [5]

Level 1: [12, 13]

Level 2: [7, 14, 2]

Level 3: [17, 23, 27, 3, 8, 11]

Code Implementation

```
import java.util.*;

class Node {
    int data;
    Node left, right;

    Node(int data) {
        this.data = data;
        left = right = null;
    }
}
```

```
public class LevelOrderTraversal {  
  
    // Function to perform Level Order Traversal  
    public static void levelOrder(Node root) {  
        if (root == null)  
            return;  
  
        Queue<Node> queue = new LinkedList<>();  
        queue.add(root);  
  
        while (!queue.isEmpty()) {  
            Node current = queue.poll();  
            System.out.print(current.data + " ");  
  
            if (current.left != null)  
                queue.add(current.left);  
            if (current.right != null)  
                queue.add(current.right);  
        }  
    }  
}
```



```
public static void main(String[] args) {  
    // Constructing a sample binary tree  
    Node root = new Node(1);  
    root.left = new Node(2);  
    root.right = new Node(3);  
    root.left.left = new Node(4);  
    root.left.right = new Node(5);  
    root.right.left = new Node(6);  
    root.right.right = new Node(7);  
  
    System.out.print("Level Order Traversal: ");  
    levelOrder(root);  
}
```

Time & Space Complexity

Traversal Type	Time Complexity	Space Complexity	Approach
Level Order (BFS)	$O(n)$	$O(n)$	Queue-based BFS

Summary

- Level Order Traversal visits all nodes level by level.
- It uses a queue to maintain the processing order (BFS).
- Ensures complete traversal of all levels before moving deeper.
- Commonly used in shortest path, serialization, and organizational hierarchy problems.

Practice Questions:

1. Binary Tree Level Order Traversal — LeetCode #102

↪ <https://leetcode.com/problems/binary-tree-level-order-traversal/>

Concept:

Perform a level-wise traversal of a binary tree using BFS and return a list of nodes at each level.

Why Practice:

- Builds understanding of queue-based traversal.
- Helps in mastering breadth-first search concepts used across tree and graph problems.

Thanks