

CSC3100 Data Structures Tutorial 7: Binary Tree Traversal

Jiawei Lian

The Chinese University of Hong Kong, Shenzhen
School of Data Science

119030043@link.cuhk.edu.cn

November 8, 2021

Content

- 1 Level order traversal from top to bottom
- 2 Level order traversal from bottom to top
- 3 Zigzag level order traversal
- 4 Coding

Outline

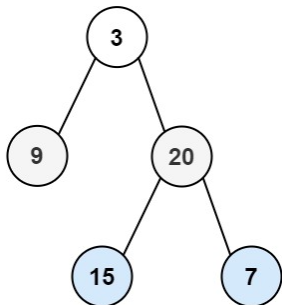
- 1 Level order traversal from top to bottom
- 2 Level order traversal from bottom to top
- 3 Zigzag level order traversal
- 4 Coding

Level order traversal from top to bottom

1 Problem statement:

Given the root of a binary tree, return the level order traversal of its nodes' values. (i.e., from left to right, level by level).

2 Example:



Input: root = [3,9,20,null,null,15,7]

Output: [[3],[9,20],[15,7]]

Example 2:

Input: root = [1]

Output: [[1]]

Example 3:

Input: root = []

Output: []

3 Constraints:

- The number of nodes in the tree is in the range [0, 2000].
- $-1000 \leq \text{Node.val} \leq 1000$

Level order traversal from top to bottom - Solution

```
public List<List<Integer>> levelOrder(Node root) {  
    List<List<Integer>> res = new ArrayList<List<Integer>>();  
    if (root == null) {  
        return res;  
    }  
  
    Queue<Node> queue = new LinkedList<Node>();  
    queue.offer(root);  
    while (!queue.isEmpty()) {  
        List<Integer> level = new ArrayList<Integer>();  
        int currentLevelSize = queue.size();  
        for (int i = 1; i <= currentLevelSize; ++i) {  
            Node node = queue.poll();  
            level.add(node.val);  
            if (node.left != null) {  
                queue.offer(node.left);  
            }  
            if (node.right != null) {  
                queue.offer(node.right);  
            }  
        }  
        res.add(level);  
    }  
  
    return res;  
}
```

Figure: Solution of example 1

Outline

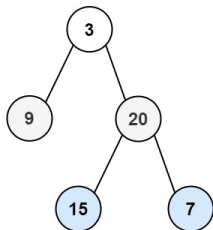
- 1 Level order traversal from top to bottom
- 2 Level order traversal from bottom to top
- 3 Zigzag level order traversal
- 4 Coding

Level order traversal from bottom to top

1 Problem statement:

Given the root of a binary tree, return the bottom-up level order traversal of its nodes' values. (i.e., from left to right, level by level from leaf to root).

2 Example:



```
Input: root = [3,9,20,null,null,15,7]
Output: [[15,7],[9,20],[3]]
```

Example 2:

```
Input: root = [1]
Output: [[1]]
```

Example 3:

```
Input: root = []
Output: []
```

3 Constraints:

- The number of nodes in the tree is in the range $[0, 2000]$.
- $-1000 \leq \text{Node.val} \leq 1000$

Level order traversal from bottom to top - Solution

```
public List<List<Integer>> levelOrderBottom(Node root) {
    List<List<Integer>> res = new LinkedList<List<Integer>>();
    if (root == null) {
        return res;
    }
    Queue<Node> queue = new LinkedList<Node>();
    queue.offer(root);
    while (!queue.isEmpty()) {
        List<Integer> level = new ArrayList<Integer>();
        int size = queue.size();
        for (int i = 0; i < size; i++) {
            Node node = queue.poll();
            level.add(node.val);
            Node left = node.left, right = node.right;
            if (left != null) {
                queue.offer(left);
            }
            if (right != null) {
                queue.offer(right);
            }
        }
        res.add(0, level);
    }
    return res;
}
```

Figure: Solution of example 2

Outline

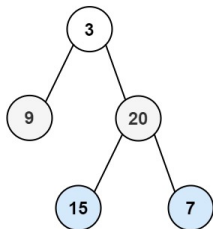
- 1 Level order traversal from top to bottom
- 2 Level order traversal from bottom to top
- 3 Zigzag level order traversal**
- 4 Coding

Zigzag level order traversal

1 Problem statement:

Given the root of a binary tree, return the zigzag level order traversal of its nodes' values. (i.e., from left to right, then right to left for the next level and alternate between).

2 Example:



Input: root = [3,9,20,null,null,15,7]
Output: [[15,7],[9,20],[3]]

Example 2:

Input: root = [1]
Output: [[1]]

Example 3:

Input: root = []
Output: []

3 Constraints:

- The number of nodes in the tree is in the range [0, 2000].
- $-1000 \leq \text{Node.val} \leq 1000$

Zigzag level order traversal - Solution

```
public List<List<Integer>> zigzagLevelOrder(Node root) {  
    List<List<Integer>> res = new LinkedList<List<Integer>>();  
    if (root == null) {  
        return res;  
    }  
  
    Queue<Node> nodeQueue = new LinkedList<Node>();  
    nodeQueue.offer(root);  
    boolean isOrderLeft = true;  
  
    while (!nodeQueue.isEmpty()) {  
        Deque<Integer> levelList = new LinkedList<Integer>();  
        int size = nodeQueue.size();  
        for (int i = 0; i < size; ++i) {  
            Node curNode = nodeQueue.poll();  
            if (isOrderLeft) {  
                levelList.offerLast(curNode.val);  
            } else {  
                levelList.offerFirst(curNode.val);  
            }  
            if (curNode.left != null) {  
                nodeQueue.offer(curNode.left);  
            }  
            if (curNode.right != null) {  
                nodeQueue.offer(curNode.right);  
            }  
        }  
        res.add(new LinkedList<Integer>(levelList));  
        isOrderLeft = !isOrderLeft;  
    }  
  
    return res;  
}
```

Figure: Solution of example 3

Outline

- 1 Level order traversal from top to bottom
- 2 Level order traversal from bottom to top
- 3 Zigzag level order traversal
- 4 Coding

Talk is cheap. Lets start coding.

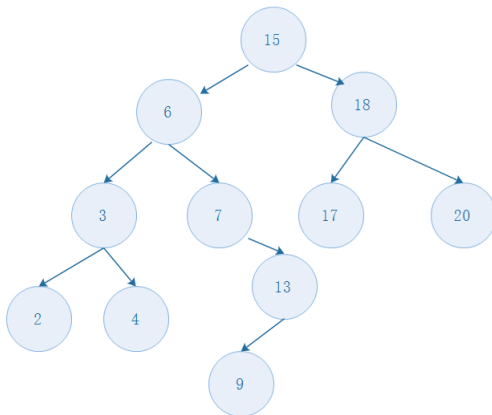


Figure: BST used in tests

References I

- <https://leetcode-cn.com/problems/binary-tree-level-order-traversal/solution/er-cha-shu-de-ceng-xu-bian-li-by-leetcode-solution/>
- <https://leetcode-cn.com/problems/binary-tree-level-order-traversal-ii/solution/er-cha-shu-de-ceng-ci-bian-li-ii-by-leetcode-solut/>
- <https://leetcode-cn.com/problems/binary-tree-zigzag-level-order-traversal/solution/er-cha-shu-de-ju-chi-xing-ceng-xu-bian-li-qsun/>

CSC3100 Data Structures Tutorial 7: Binary Tree Traversal

Jiawei Lian

The Chinese University of Hong Kong, Shenzhen
School of Data Science

`119030043@link.cuhk.edu.cn`

November 8, 2021