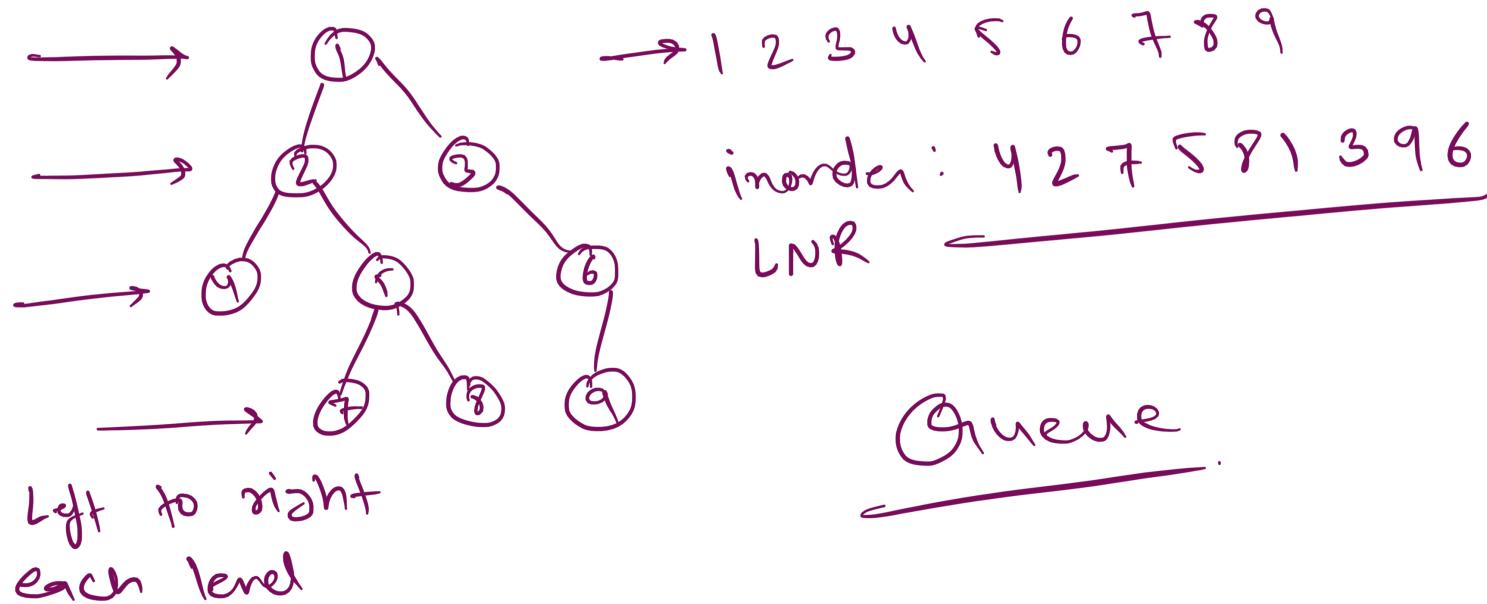


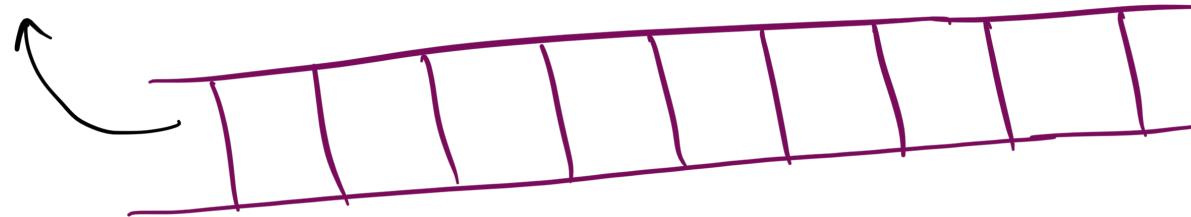
Binary Tree - I



LeetCode - 102

Print Level-Order Traversal of a Binary Tree



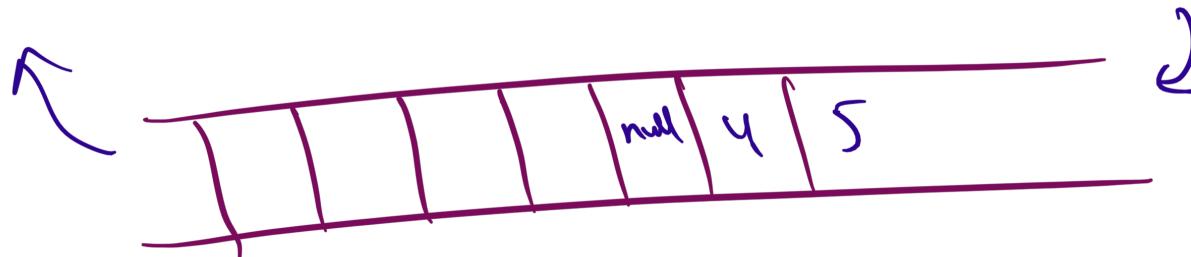


1, 2, 3, 4, 5, 6, 7, 8, 9

while(!queue.isEmpty())

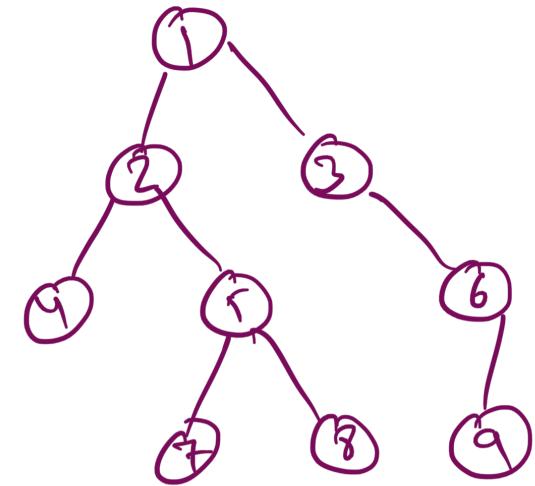
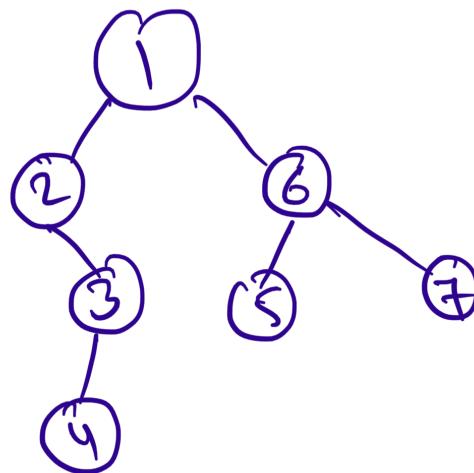
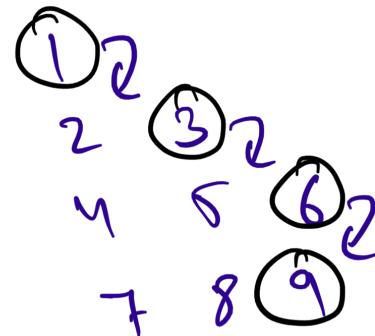
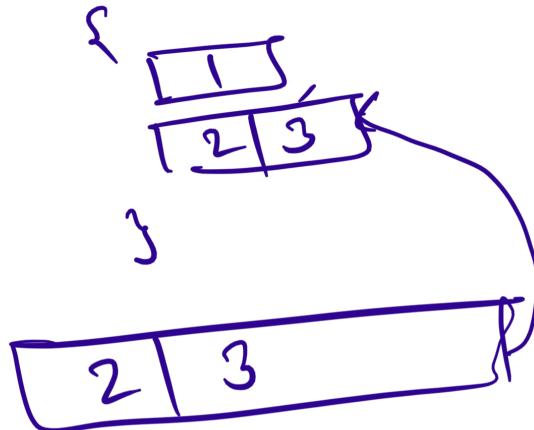
1. remove from queue
2. Insert left and right into queue

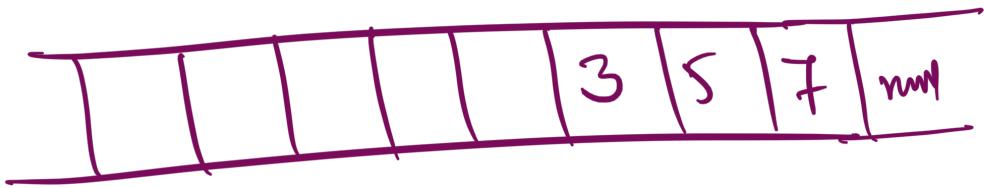
Queue < Node >



1 2
2 3 2
4 5 6 2
7 8 9 2

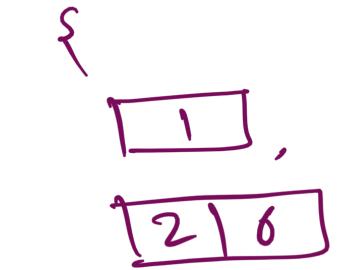
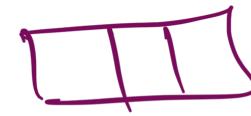
similar



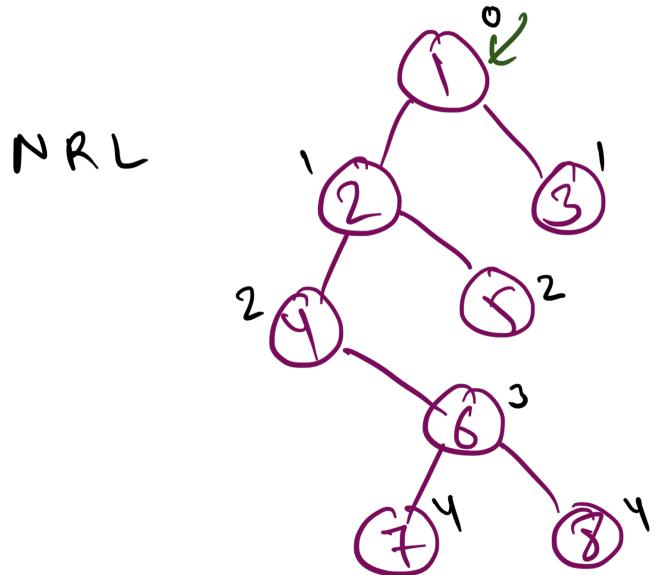


curr = null

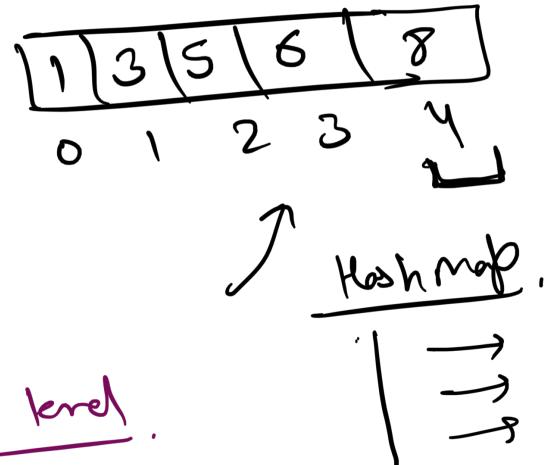
//

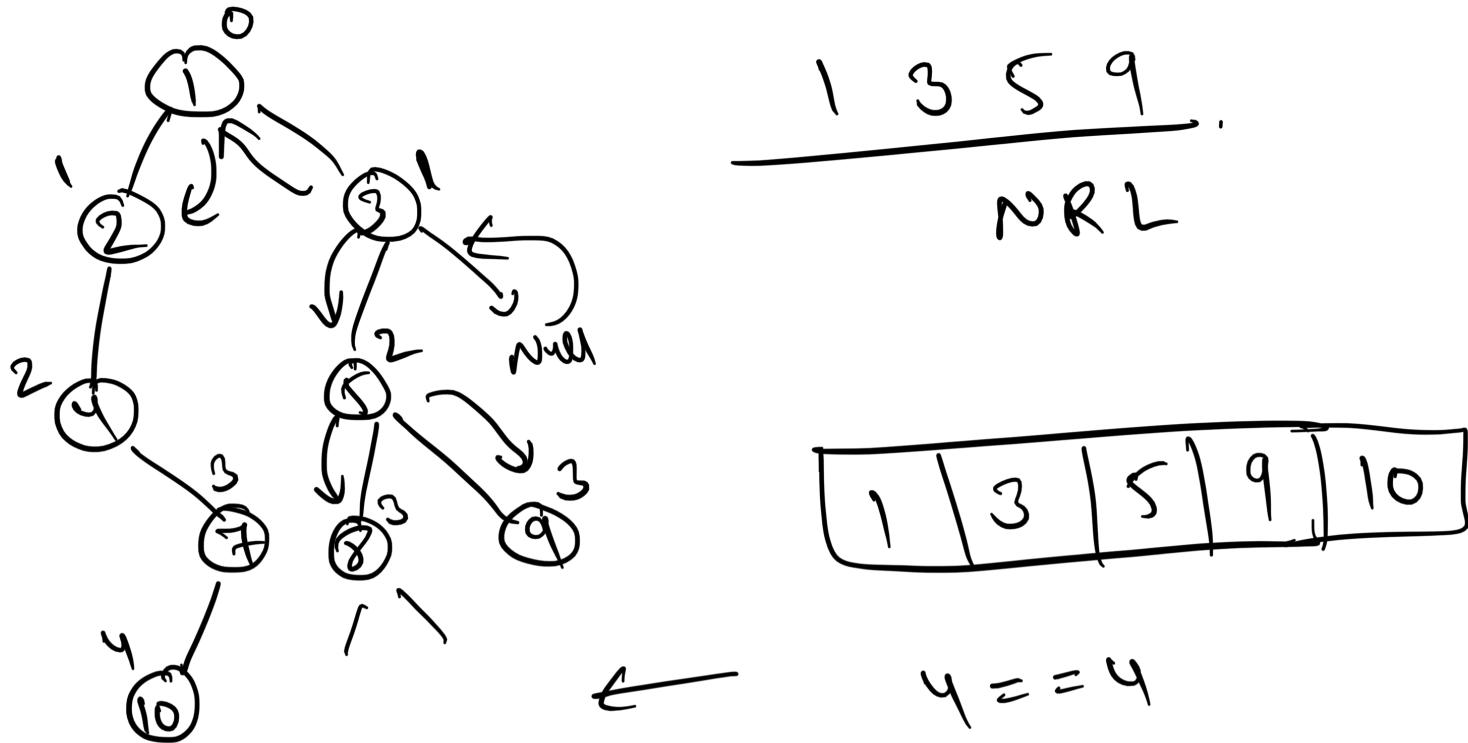


Print Right-view of a Binary Tree



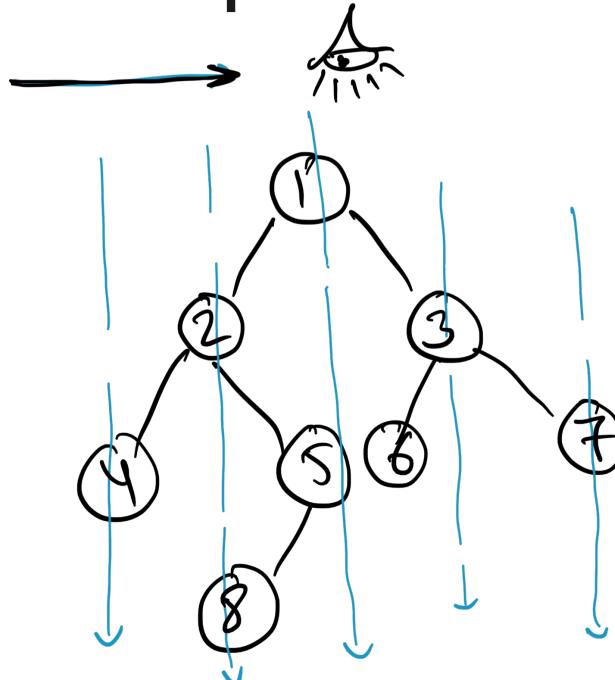
Print the last element of each level





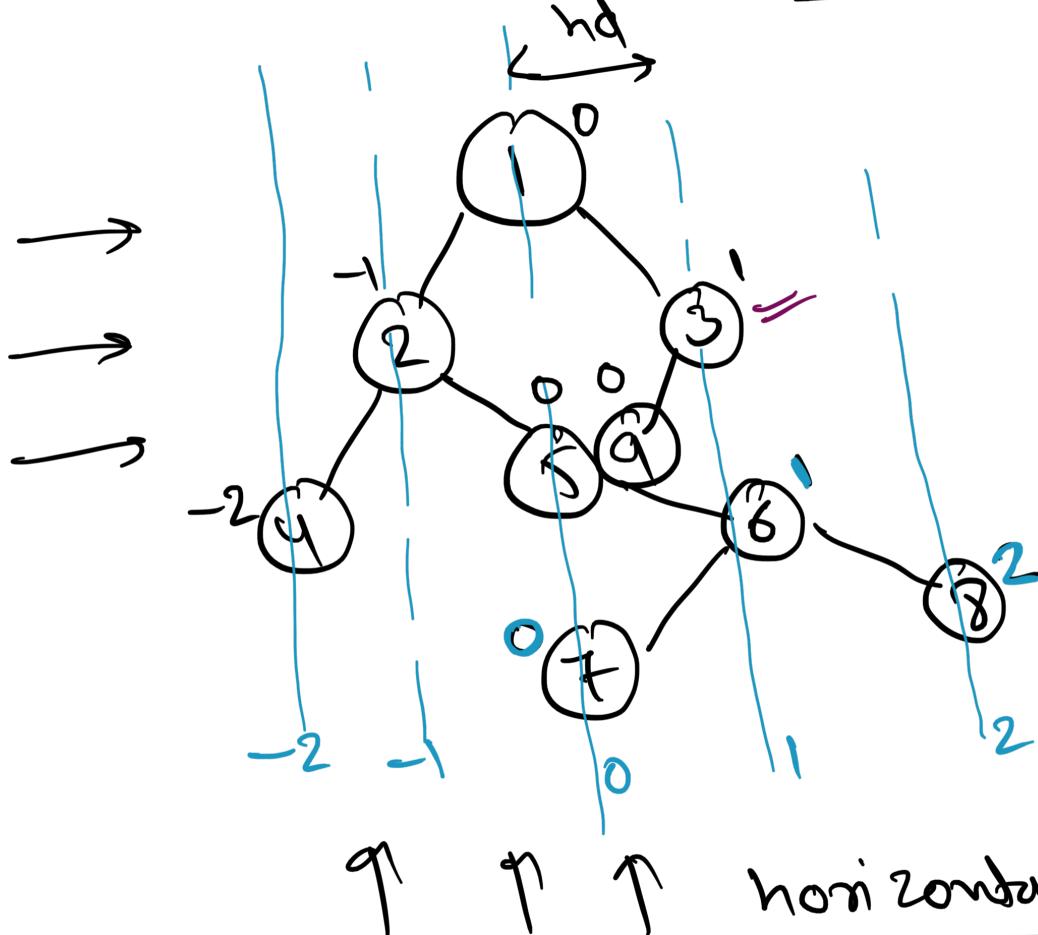
#

Print Top View of a Binary Tree

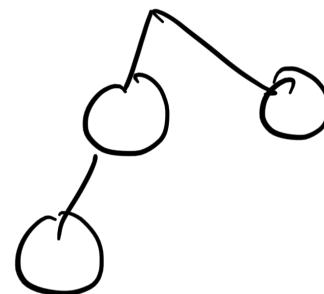


→ 4, 2, 1, 3, 7.
→ 4, 2, 8, 1, 5, 6, 3, 7

Vertical order Traversal



4 2 1 5 9 7 3 6 8



horizontal distance → Recursion X (DFS)
→ Queue (BFS)

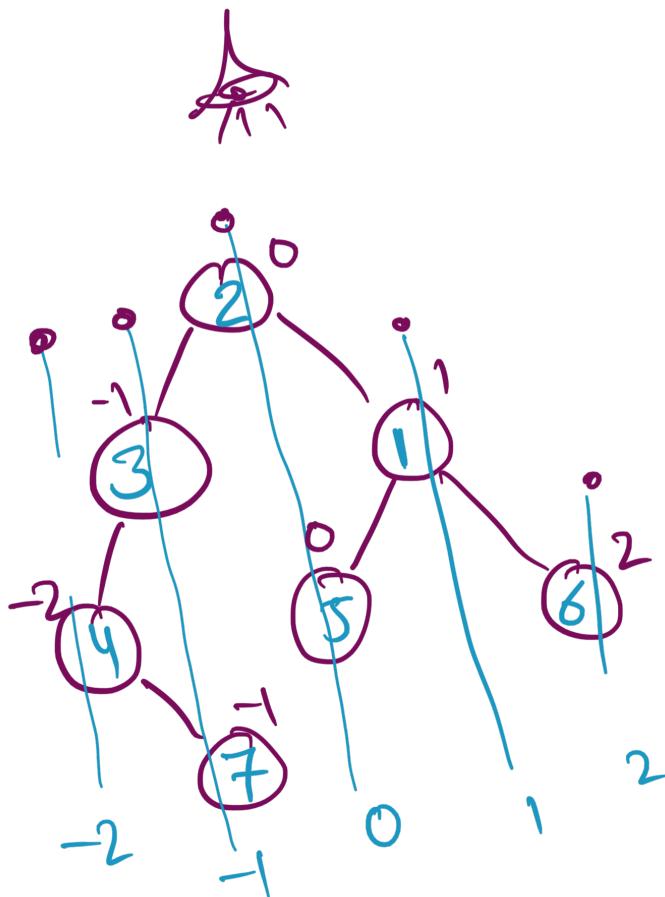
(hd, Node)

-2	→	4	1				
-1	→	2					
0	→	1	5	9	5	7	
1	→	3	6				
2	→	8					

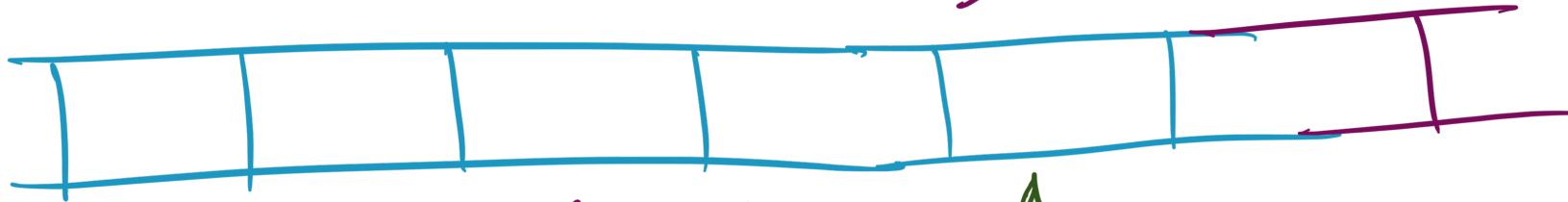
Curr = (2, 8)

3
TreeMap<Integer, ArrayList>

-2	→	4	1	
-1	→	3	7	
0	→	2	5	
1	→	1		
2	→	6		



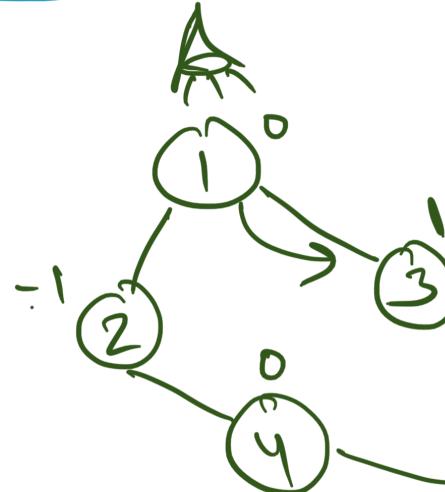
ans [4 | 3 | 7 | 2 | 5 | 1 | 6] //



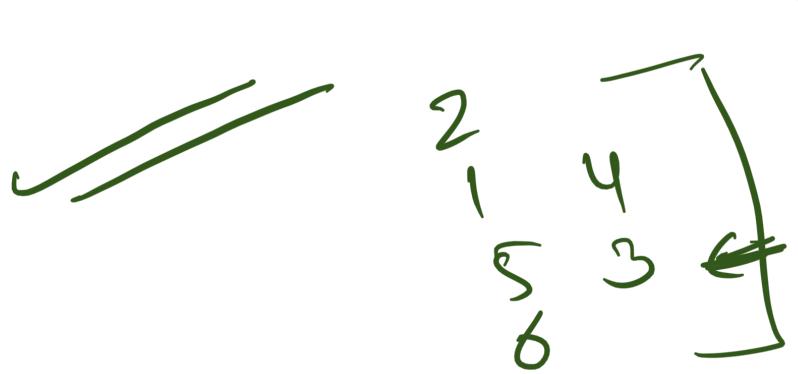
cur = (-1, 7)

NLR

0 → 1
-1 → 2
1 → 5 X
2 → 6



2 1 3 6



Practice Problems

1. Print the Zig-Zag order Traversal of a Binary Tree
2. Print the Left View of a Binary Tree ✓
3. Print the Bottom View of a Binary Tree
4. Modified Vertical order traversal:
<https://leetcode.com/problems/vertical-order-traversal-of-a-binary-tree/>
5. Cousins in a Binary Tree: <https://leetcode.com/problems/cousins-in-binary-tree/>
6. Populate the Next Pointer in Each Node:
<https://leetcode.com/problems/populating-next-right-pointers-in-each-node/>