

Week 12- Day 2 : Coding Challenge

(Maximum marks -15)

Q-1) Print vertical order traversal, or Top view of a binary tree

<https://leetcode.com/problems/vertical-order-traversal-of-a-binary-tree/>

(5 marks)

(Easy)

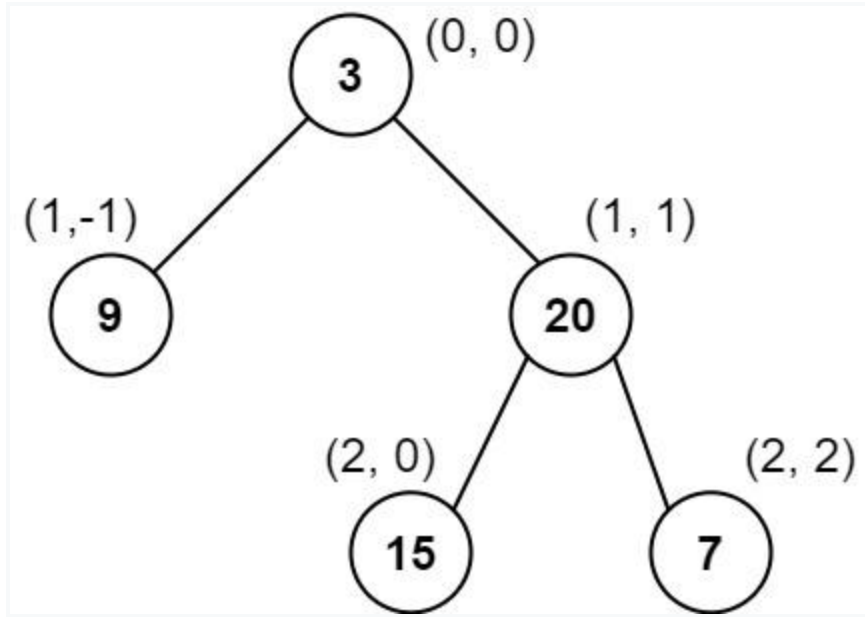
Given the root of a binary tree, calculate the vertical order traversal of the binary tree.

For each node at position (row, col), its left and right children will be at positions (row + 1, col - 1) and (row + 1, col + 1) respectively. The root of the tree is at (0, 0).

The vertical order traversal of a binary tree is a list of top-to-bottom orderings for each column index starting from the leftmost column and ending on the rightmost column. There may be multiple nodes in the same row and same column. In such a case, sort these nodes by their values.

Return *the vertical order traversal of the binary tree*.

Example 1:



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

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

Explanation:

Column -1: Only node 9 is in this column.

Column 0: Nodes 3 and 15 are in this column in that order from top to bottom.

Column 1: Only node 20 is in this column.

Column 2: Only node 7 is in this column.

Q-2)Sum of Root To Leaf Binary Numbers

(5 marks)

<https://leetcode.com/problems/sum-of-root-to-leaf-binary-numbers/>

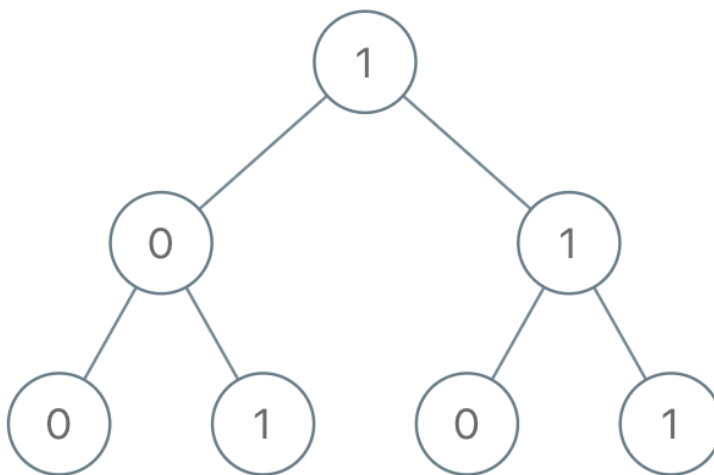
(Easy)

You are given the root of a binary tree where each node has a value 0 or 1. Each root-to-leaf path represents a binary number starting with the most significant bit. For example, if the path is 0 -> 1 -> 1 -> 0 -> 1, then this could represent 01101 in binary, which is 13.

For all leaves in the tree, consider the numbers represented by the path from the root to that leaf.

Return *the sum of these numbers*. The answer is guaranteed to fit in a 32-bits integer.

Example 1:



Input: root = [1,0,1,0,1,0,1]

Output: 22

Explanation: (100) + (101) + (110) + (111) = 4 + 5 + 6 + 7 = 22

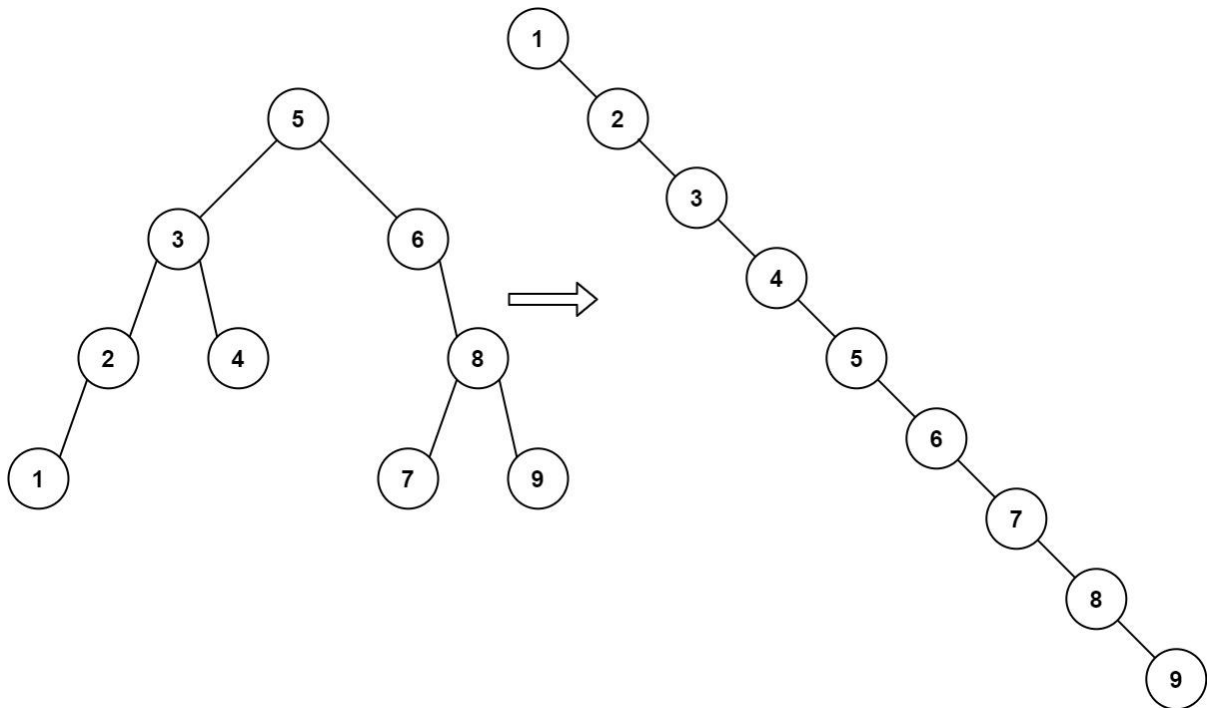
Q-3)Increasing Order Search Tree (5 marks)

<https://leetcode.com/problems/increasing-order-search-tree/>

(Easy)

Given the **root** of a binary search tree, rearrange the tree in in-order so that the leftmost node in the tree is now the root of the tree, and every node has no left child and only one right child.

Example 1:



Input: root = [5,3,6,2,4,null,8,1,null,null,null,7,9]

Output: [1,null,2,null,3,null,4,null,5,null,6,null,7,null,8,null,9]

Marks distribution:

Question 1,2 and 3 carry 5 marks each.