Approach1

- The first element of the Level order Traversal is necessarily the root of the binary tree.
- The nodes which appear left of the root in the inorder traversal are necessarily in the left subtree, and those which appear in the right are in the right subtree of the right subtree.
- We can make use of the fact that the nodes of a subtree of the binary tree will form a continuous segment in the inorder array.
- We can maintain a queue with each element in a queue containing three integers, namely the height of the current subtree, the starting index, and the ending index of the subtree in the inorder array.
- Then the algorithm is:
    - We will initially push (0, 0, 'N' - 1) in the queue.
    - We will start traversing the level order array.
    - When we are at index 'i' in the level order array we will pop out the front element of the queue.
        1. Let that popped-out element be ('HEIGHT', 'LEFTINDEX', 'RIGHTINDEX').
        2. We will start iterating from the 'LEFTINDEX' to 'RIGHTINDEX' of the inorder array and find the index of element 'LEVELORDER[i]' in the inorder array. Let that index be 'j'.
        3. If ('j' - 1) is greater than equal to leftIndex, we will push ('HEIGHT' + 1, 'LEFTINDEX', 'j' - 1) into the queue.
        4. Similarly, if ('j' + 1) is less than equal to rightIndex we will push ('HEIGHT' + 1, 'j' + 1, 'RIGHTINDEX') into the queue.
    - The maximum height encountered in this process will be the height of the binary tree.

APPROACH 2

- The N^2 approach can be further optimized by storing the position of node 'i' in the inorder array so that we can save time on searching for the position.
- We can declare an auxiliary array 'POS' of size 'N' with each element storing the index of node 'i', so that we can get the position of a node in an inorder array in constant time.
- We can follow the same previous algorithm with one modification-
    - Instead of iterating from 'LEFTINDEX' to 'RIGHTINDEX' to find the position of 'LEVELORDER[i]', we can directly query the required position from the 'POS' array.