https://leetcode.com/problems/count-complete-tree-nodes/description/

Complete Binary Trees are filled from left to right "strictly".

So left node might(or equal) be having 1 node greater than right node.

Check method "countNodes\_TimeLimitExceeded" which fails.

It is a simple height of tree with "return as 0 if null" and without (Math.max)

===log(n) solution approach=======

1) Key point is since data is strictly filled from left. I can verify left height and right height.

If both are same. Then I got my output. (2 power height -1). Ex: "1 level 1". "2 level 4-1", "3 level 8-1", "4 level 16-1".

2) Else do step1 for left and right.

3) Save the Height of Left and Right

===========Note======

1) When the tree is properly balanced and tree is big. The logic will be very efficient.

2) Worst case it has to visit most node. Sometimes even more than n nodes for smaller tree.

Check diagram "CountCompleteTreeNodes\_WorstCase.jpg" and "CountCompleteTreeNodes\_BestCase.jpg" in US Problems