A *complete* binary tree is a binary tree in which every level, except possibly the last, is completely filled, and all nodes are as far left as possible.

Write a data structure CBTInserter that is initialized with a complete binary tree and supports the following operations:

* CBTInserter(TreeNode root) initializes the data structure on a given tree with head node root;
* CBTInserter.insert(int v) will insert a TreeNode into the tree with value node.val = v so that the tree remains complete, **and returns the value of the parent of the inserted TreeNode**;
* CBTInserter.get\_root() will return the head node of the tree.

**Example 1:**

**Input:** inputs = ["CBTInserter","insert","get\_root"], inputs = [[[1]],[2],[]]

**Output:** [null,1,[1,2]]

**Example 2:**

**Input:** inputs = ["CBTInserter","insert","insert","get\_root"], inputs = [[[1,2,3,4,5,6]],[7],[8],[]]

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

**Note:**

1. The initial given tree is complete and contains between 1 and 1000 nodes.
2. CBTInserter.insert is called at most 10000 times per test case.
3. Every value of a given or inserted node is between 0 and 5000.