<https://www.geeksforgeeks.org/create-a-mirror-tree-from-the-given-binary-tree/>

A screenshot of a computer

Description automatically generated

**Approach:** Write a recursive function that will take two nodes as the argument, one of the original tree and the other of the newly created tree. Now, for every passed node of the original tree, create a corresponding node in the mirror tree and then recursively call the same method for the child nodes but passing the left child of the original tree node with the right child of the mirror tree node and the right child of the original tree node with the left child of the mirror tree node.

**Time Complexity:** O(n), where n is the number of nodes in the tree. This is because we need to visit each node in the tree exactly once to swap its left and right child nodes.  
**Auxiliary Space:** O(h), where h is the height of the binary tree. This is because the maximum amount of space used by the algorithm at any given time is the size of the call stack, which is at most equal to the height of the binary tree. This is because each recursive call to mirror adds a new frame to the call stack, and the stack grows to a maximum depth of the height of the tree. Therefore, the space used by the algorithm is proportional to the height of the tree.

A screenshot of a computer program

Description automatically generated A screenshot of a computer program

Description automatically generated

**Approach 2:**  
 In order to change the original tree in its mirror tree, then we simply swap the left and right link of each node. If the node is leaf node then do nothing.

A screenshot of a computer code

Description automatically generated

