```python
# Copyright © 2020 AlgoExpert, LLC. All rights reserved.

class BinaryTree:
    def __init__(self, value, left=None, right=None):
        self.value = value
        self.left = left
        self.right = right


# O(n) time | O(d) space - where n is the number of nodes in the Binary Tree
# and d is the depth (height) of the Binary Tree
def flattenBinaryTree(root):
    leftMost, _ = flattenTree(root)
    return leftMost


def flattenTree(node):
    if node.left is None:
        leftMost = node
    else:
        leftSubtreeLeftMost, leftSubtreeRightMost = flattenTree(node.left)
        connectNodes(leftSubtreeRightMost, node)
        leftMost = leftSubtreeLeftMost

    if node.right is None:
        rightMost = node
    else:
        rightSubtreeLeftMost, rightSubtreeRightMost = flattenTree(node.right)
        connectNodes(node, rightSubtreeLeftMost)
        rightMost = rightSubtreeRightMost

    return [leftMost, rightMost]


def connectNodes(left, right):
    left.right = right
    right.left = left

```