```
1  // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3  using System.Collections.Generic;
4
5  public class Program {
6    // O(n) time | O(n) space - where n is the number of nodes in the Binary Tree
7    public static BinaryTree FlattenBinaryTree(BinaryTree root) {
8      List<BinaryTree> inOrderNodes = getNodesInOrder(root, new List<BinaryTree>());
9      for (int i = 0; i < inOrderNodes.Count - 1; i++) {
10       BinaryTree leftNode = inOrderNodes[i];
11       BinaryTree rightNode = inOrderNodes[i + 1];
12       leftNode.right = rightNode;
13       rightNode.left = leftNode;
14     }
15     return inOrderNodes[0];
16   }
17
18   public static List<BinaryTree> getNodesInOrder(BinaryTree tree, List<BinaryTree> array) {
19     if (tree != null) {
20       getNodesInOrder(tree.left, array);
21       array.Add(tree);
22       getNodesInOrder(tree.right, array);
23     }
24     return array;
25   }
26
27   public class BinaryTree {
28     public int value;
29     public BinaryTree left = null;
30     public BinaryTree right = null;
31
32     public BinaryTree(int value) {
33       this.value = value;
34     }
35   }
36 }
37
```