```
1  // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3  class BinaryTree {
4    constructor(value) {
5      this.value = value;
6      this.left = null;
7      this.right = null;
8    }
9  }
10
11 // O(n) time | O(n) space - where n is the number of nodes in the Binary Tree
12 function flattenBinaryTree(root) {
13   const inOrderNodes = getNodesInOrder(root, []);
14   for (let i = 0; i < inOrderNodes.length - 1; i++) {
15     const leftNode = inOrderNodes[i];
16     const rightNode = inOrderNodes[i + 1];
17     leftNode.right = rightNode;
18     rightNode.left = leftNode;
19   }
20   return inOrderNodes[0];
21 }
22
23 function getNodesInOrder(tree, array) {
24   if (tree !== null) {
25     getNodesInOrder(tree.left, array);
26     array.push(tree);
27     getNodesInOrder(tree.right, array);
28   }
29   return array;
30 }
31
32 exports.BinaryTree = BinaryTree;
33 exports.flattenBinaryTree = flattenBinaryTree;
34
```