```cpp
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

#include <vector>
using namespace std;

class BinaryTree {
public:
  int value;
  BinaryTree *left = NULL;
  BinaryTree *right = NULL;

  BinaryTree(int value);
};

vector<BinaryTree *> getNodesInOrder(BinaryTree *tree,
                                     vector<BinaryTree *> *array);

// O(n) time | O(n) space - where n is the number of nodes in the Binary Tree
BinaryTree *flattenBinaryTree(BinaryTree *root) {
  vector<BinaryTree *> inOrderNodes =
      getNodesInOrder(root, new vector<BinaryTree *>{});
  for (int i = 0; i < inOrderNodes.size() - 1; i++) {
    BinaryTree *leftNode = inOrderNodes[i];
    BinaryTree *rightNode = inOrderNodes[i + 1];
    leftNode->right = rightNode;
    rightNode->left = leftNode;
  }
  return inOrderNodes[0];
}

vector<BinaryTree *> getNodesInOrder(BinaryTree *tree,
                                     vector<BinaryTree *> *array) {
  if (tree != NULL) {
    getNodesInOrder(tree->left, array);
    array->push_back(tree);
    getNodesInOrder(tree->right, array);
  }
  return *array;
}

```