Solution 1

```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);
};

void mutate(BinaryTree *node, BinaryTree *parent, bool isLeftChild);

// 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
BinaryTree *rightSiblingTree(BinaryTree *root) {
  mutate(root, NULL, false);
  return root;
}

void mutate(BinaryTree *node, BinaryTree *parent, bool isLeftChild) {
  if (node == NULL)
    return;

  auto left = node->left;
  auto right = node->right;
  mutate(left, node, true);
  if (parent == NULL) {
    node->right = NULL;
  } else if (isLeftChild) {
    node->right = parent->right;
  } else {
    if (parent->right == NULL) {
      node->right = NULL;
    } else {
      node->right = parent->right->left;
    }
  }
  mutate(right, node, false);
}

```