Our Solution(s) Run Code

Solution 1 Solution 2

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

// O(n) time | O(d) space
function invertBinaryTree(tree) {
  if (tree === null) return;
  swapLeftAndRight(tree);
  invertBinaryTree(tree.left);
  invertBinaryTree(tree.right);
}

function swapLeftAndRight(tree) {
  const left = tree.left;
  tree.left = tree.right;
  tree.right = left;
}

exports.invertBinaryTree = invertBinaryTree;

```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```
function invertBinaryTree(tree) {
  // Write your code here.
}

// Do not edit the line below.
exports.invertBinaryTree = invertBinaryTree;

```

Our Tests

Custom Output Submit Code

```
const chai = require('chai');

class BinaryTree {
  constructor(value) {
    this.value = value;
    this.left = null;
    this.right = null;
  }

  insertValues(...values) {
    if (0 == values.length) return;
    const queue = [this];
    while (queue.length > 0) {
      let current = queue.shift();
      if (current.left === null) {
        current.left = new BinaryTree(values[0]);
        break;
      }
      queue.push(current.left);
      if (current.right === null) {
        current.right = new BinaryTree(values[0]);
```

Run or submit code when you're ready.