Our Solution(s)   Run Code

Solution 1   Solution 2

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

// O(n) time | O(n) space
function invertBinaryTree(tree) {
  const queue = [tree];
  while (queue.length) {
    const current = queue.shift();
    if (current === null) continue;
    swapLeftAndRight(current);
    queue.push(current.left);
    queue.push(current.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

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

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

Our Solution(s)   Run Code

Custom Output   Raw Output   Submit Code

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

// O(n) time | O(n) space
function invertBinaryTree(tree) {
  const queue = [tree];
  while (queue.length) {
    const current = queue.shift();
    if (current === null) continue;
    swapLeftAndRight(current);
```

Run or submit code when you're ready.