Our Solution(s)   Run Code

Solution 1   Solution 2

```go
1  // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3  package main
4
5  type BinaryTree struct {
6    Value int
7
8    Left  *BinaryTree
9    Right *BinaryTree
10 }
11
12 // O(n) time | O(n) space
13 func (tree *BinaryTree) InvertBinaryTree() {
14   queue := []*BinaryTree{tree}
15   for len(queue) > 0 {
16     current := queue[0]
17     queue = queue[1:]
18     if current == nil {
19       continue
20     }
21     current.Left, current.Right = current.Right, current.Left
22     queue = append(queue, current.Left, current.Right)
23   }
24 }
25
```

Your Solutions   Run Code

Solution 1   Solution 2   Solution 3

```go
1  package main
2
3  type BinaryTree struct {
4    Value int
5
6    Left  *BinaryTree
7    Right *BinaryTree
8  }
9
10 func (tree *BinaryTree) InvertBinaryTree() {
11   // Write your code here.
12 }
13
```

Custom Output   **Raw Output**   Submit Code

Run or submit code when you're ready.