Our Solution(s) Run Code

Solution 1 Solution 2

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

package main

type BinaryTree struct {
  Value int

  Left  *BinaryTree
  Right *BinaryTree
}

// O(n) time | O(n) space
func (tree *BinaryTree) InvertBinaryTree() {
  queue := []*BinaryTree{tree}
  for len(queue) > 0 {
    current := queue[0]
    queue = queue[1:]
    if current == nil {
      continue
    }
    current.Left, current.Right = current.Right, current.Left
    queue = append(queue, current.Left, current.Right)
  }
}

```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```go
package main

type BinaryTree struct {
  Value int

  Left  *BinaryTree
  Right *BinaryTree
}

func (tree *BinaryTree) InvertBinaryTree() {
  // Write your code here.
}

```

Our Tests

Custom Output Submit Code

```
package

import (
	"github.com/structure/heaviliy/template"
)

func NewBinaryTree(root int, values ...int) *BinaryTree {
	tree := &BinaryTree{value: root}
	for _, value := range values {
		tree.InsertValue(value)
	}
	return tree
}

func (tree *BinaryTree) InsertValue(val int) *BinaryTree {
	queue := []*BinaryTree{tree}
	for len(queue) > 0 {
		current := queue[0]
		queue = queue[1:]
		if current.left == nil {
			current.left = NewBinaryTree(value)
			break
```

Run or submit code when you're ready.