AlgoExpert

Quad Layout | Go | 12px | Sublime | Monokai | 00:00:00

Prompt | Scratchpad | Our Solution(s) | Video Explanation | 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 - where n is the number of nodes
// in the Binary Tree
func FlattenBinaryTree(root *BinaryTree) *BinaryTree {
  inOrderNodes := []*BinaryTree{}
  getNodesInOrder(root, &inOrderNodes)
  for i := 0; i < len(inOrderNodes)-1; i++ {
    leftNode := inOrderNodes[i]
    rightNode := inOrderNodes[i+1]
    leftNode.Right = rightNode
    rightNode.Left = leftNode
  }
  return inOrderNodes[0]
}

func getNodesInOrder(tree *BinaryTree, array *[]*BinaryTree) {
  if tree != nil {
    getNodesInOrder(tree.Left, array)
    *array = append(*array, tree)
    getNodesInOrder(tree.Right, array)
  }
}

```