AlgoExpert　Quad Layout　Go　14px　Sublime　Monokai　00:00:

**Our Solution(s)**　Run Code

**Solution 1**

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

package main

type BinaryTree struct {
  Value int

  Left   *BinaryTree
  Right  *BinaryTree
  Parent *BinaryTree
}

// O(n) time | O(1) space
func (tree *BinaryTree) IterativeInOrderTraversal(callback func(int)) {
  var previous, next *BinaryTree
  current := tree
  for current != nil {
    if previous == nil || previous == current.Parent {
      if current.Left != nil {
        next = current.Left
      } else {
        callback(current.Value)
        if current.Right != nil {
          next = current.Right
        } else {
          next = current.Parent
        }
      }
    } else if previous == current.Left {
      callback(current.Value)
      if current.Right != nil {
        next = current.Right
      } else {
```

**Your Solutions**　Run Code

**Solution 1**　**Solution 2**　**Solution 3**

```go
package main

type BinaryTree struct {
  Value int

  Left   *BinaryTree
  Right  *BinaryTree
  Parent *BinaryTree
}

func (tree *BinaryTree) IterativeInOrderTraversal(callback func(int)) {
  // Write your code here.
}
```

**Our Tests**

[illegible]

**Custom Output**　Submit Code

```
  tree := &BinaryTree{Value: root}
  for _, value := range values {
    tree.Insert(value)
  }
  return tree
}

func (tree *BinaryTree) Insert(value int) *BinaryTree {
  queue := []*BinaryTree{tree}
  for len(queue) > 0 {
    current := queue[0]
    queue = queue[1:]
    if current.Left == nil {
      current.Left = NewBinaryTree(value)
      current.Left.Parent = current
```

**Run or submit code when you're ready.**