Solution 1

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

class Program {
    class BinaryTree {
        var value: Int
        var parent: BinaryTree?
        var left: BinaryTree?
        var right: BinaryTree?

        init(value: Int, parent: BinaryTree?) {
            self.value = value
            self.parent = parent
        }
    }

    // O(n) time | O(1) space
    func iterativeInOrderTraversal(_ tree: BinaryTree, _ callback: (BinaryTree) -> Void) {
        var previousNode: BinaryTree?
        var currentNode: BinaryTree? = tree

        while currentNode !== nil {
            let nextNode: BinaryTree?
            if previousNode === nil || previousNode === currentNode?.parent {
                if currentNode?.left !== nil {
                    nextNode = currentNode?.left
                } else {
                    callback(currentNode!)
                    nextNode = currentNode?.right !== nil ? currentNode?.right : currentNode?.parent
                }
            } else if previousNode === currentNode?.left {
                callback(currentNode!)
                nextNode = currentNode?.right !== nil ? currentNode?.right : currentNode?.parent
            } else {
                nextNode = currentNode?.parent
            }

            previousNode = currentNode
            currentNode = nextNode
        }
    }
}

```