Our Solution(s) Run Code

Solution 1

```python
# Copyright © 2020 AlgoExpert, LLC. All rights reserved.

# O(n) time | O(1) space
def iterativeInOrderTraversal(tree, callback):
    previousNode = None
    currentNode = tree
    while currentNode is not None:
        if previousNode is None or previousNode == currentNode.parent:
            if currentNode.left is not None:
                nextNode = currentNode.left
            else:
                callback(currentNode)
                nextNode = currentNode.right if currentNode.right is n
        elif previousNode == currentNode.left:
            callback(currentNode)
            nextNode = currentNode.right if currentNode.right is not N
        else:
            nextNode = currentNode.parent
        previousNode = currentNode
        currentNode = nextNode

```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```python
def iterativeInOrderTraversal(tree, callback):
    # Write your code here.
    pass

```

Our Tests

Custom Output Submit Code

```
        self.left = None
        self.right = None
        self.parent = parent

    def insert(self, values, i=0):
        if i >= len(values):
            return
        queue = [self]
        while len(queue) > 0:
            current = queue.pop(0)
            if current.left is None:
                current.left = BinaryTree(values[i], current)
                break
            queue.append(current.left)
            if current.right is None:
```

Run or submit code when you're ready.