Our Solution(s) Run Code

Solution 1

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

import java.util.function.Function;

class Program {
  // O(n) time | O(1) space
  public static void iterativeInOrderTraversal(
      BinaryTree tree, Function<BinaryTree, Void> callback) {
    BinaryTree previousNode = null;
    BinaryTree currentNode = tree;
    while (currentNode != null) {
      BinaryTree nextNode;
      if (previousNode == null || previousNode == currentNode.parent)
        if (currentNode.left != null) {
          nextNode = currentNode.left;
        } else {
          callback.apply(currentNode);
          nextNode = currentNode.right != null ? currentNode.right : c
        }
      } else if (previousNode == currentNode.left) {
        callback.apply(currentNode);
        nextNode = currentNode.right != null ? currentNode.right : cur
      } else {
        nextNode = currentNode.parent;
      }
      previousNode = currentNode;
      currentNode = nextNode;
    }
  }

  static class BinaryTree {
    public int value;
    public BinaryTree left;
```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```java
import java.util.function.Function;

class Program {
  public static void iterativeInOrderTraversal(
      BinaryTree tree, Function<BinaryTree, Void> callback) {
    // Write your code here.
  }

  static class BinaryTree {
    public int value;
    public BinaryTree left;
    public BinaryTree right;
    public BinaryTree parent;

    public BinaryTree(int value) {
      this.value = value;
    }

    public BinaryTree(int value, BinaryTree parent) {
      this.value = value;
      this.parent = parent;
    }
  }
}

```

Our Tests

Custom Output Submit Code

Run or submit code when you're ready.