Our Solution(s) Run Code

Solution 1

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

package main

type BST struct {
  Value int

  Left  *BST
  Right *BST
}

// O(n) time | O(n) space
func (tree *BST) InOrderTraverse(array []int) []int {
  if tree.Left != nil {
    array = tree.Left.InOrderTraverse(array)
  }
  array = append(array, tree.Value)
  if tree.Right != nil {
    array = tree.Right.InOrderTraverse(array)
  }
  return array
}

// O(n) time | O(n) space
func (tree *BST) PreOrderTraverse(array []int) []int {
  array = append(array, tree.Value)
  if tree.Left != nil {
    array = tree.Left.PreOrderTraverse(array)
  }
  if tree.Right != nil {
    array = tree.Right.PreOrderTraverse(array)
  }
  return array
}

// O(n) time | O(n) space
func (tree *BST) PostOrderTraverse(array []int) []int {
  if tree.Left != nil {
    array = tree.Left.PostOrderTraverse(array)
  }
  if tree.Right != nil {
    array = tree.Right.PostOrderTraverse(array)
  }
  array = append(array, tree.Value)
  return array
}

```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```go
package main

type BST struct {
  Value int

  Left  *BST
  Right *BST
}

func (tree *BST) InOrderTraverse(array []int) []int {
  // Write your code here.
  return nil
}

func (tree *BST) PreOrderTraverse(array []int) []int {
  // Write your code here.
  return nil
}

func (tree *BST) PostOrderTraverse(array []int) []int {
  // Write your code here.
  return nil
}

```

Custom Output Raw Output Submit Code

Run or submit code when you're ready.