

Our Solution(s)

Run Code

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 package main
4
5 type BST struct {
6     Value int
7
8     Left *BST
9     Right *BST
10 }
11
12 // O(n) time | O(n) space
13 func (tree *BST) InOrderTraverse(array []int) []int {
14     if tree.Left != nil {
15         array = tree.Left.InOrderTraverse(array)
16     }
17     array = append(array, tree.Value)
18     if tree.Right != nil {
19         array = tree.Right.InOrderTraverse(array)
20     }
21     return array
22 }
23
24 // O(n) time | O(n) space
25 func (tree *BST) PreOrderTraverse(array []int) []int {
26     array = append(array, tree.Value)
27     if tree.Left != nil {
28         array = tree.Left.PreOrderTraverse(array)
29     }
30     if tree.Right != nil {
31         array = tree.Right.PreOrderTraverse(array)
32     }
33     return array
```

Your Solutions

Run Code

Solution 1   Solution 2   Solution 3

```
1 package main
2
3 type BST struct {
4     Value int
5
6     Left *BST
7     Right *BST
8 }
9
10 func (tree *BST) InOrderTraverse(array []int) []int {
11     // Write your code here.
12     return nil
13 }
14
15 func (tree *BST) PreOrderTraverse(array []int) []int {
16     // Write your code here.
17     return nil
18 }
19
20 func (tree *BST) PostOrderTraverse(array []int) []int {
21     // Write your code here.
22     return nil
23 }
24
```

Run or submit code when you're ready.