

Our Solution(s)Run Code

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class Program {
4     class BST {
5         var value: Int?
6         var left: BST?
7         var right: BST?
8
9         init(value: Int) {
10             self.value = value
11             left = nil
12             right = nil
13         }
14     }
15
16     // O(n) time | O(n) space
17     func inorderTraversal(tree: BST?, array: inout [Int]) -> [Int] {
18         if tree != nil {
19             inorderTraversal(tree: tree?.left, array: &array)
20
21             if let value = tree?.value {
22                 array.append(value)
23             }
24
25             inorderTraversal(tree: tree?.right, array: &array)
26         }
27
28         return array
29     }
30
31     // O(n) time | O(n) space
32     func preorderTraversal(tree: BST?, array: inout [Int]) -> [Int] {
33         if tree != nil {
```

Your SolutionsRun Code

Solution 1Solution 2Solution 3

```
1 class Program {
2     // This is an input class. Do not edit.
3     class BST {
4         var value: Int?
5         var left: BST?
6         var right: BST?
7
8         init(value: Int) {
9             self.value = value
10            left = nil
11            right = nil
12        }
13    }
14
15    func inorderTraversal(tree: BST?, array: inout [Int]) -> [Int] {
16        // Write your code here.
17        return []
18    }
19
20    func preorderTraversal(tree: BST?, array: inout [Int]) -> [Int] {
21        // Write your code here.
22        return []
23    }
24
25    func postOrderTraversal(tree: BST?, array: inout [Int]) -> [Int] {
26        // Write your code here.
27        return []
28    }
29 }
30
```

