Our Solution(s) Run Code

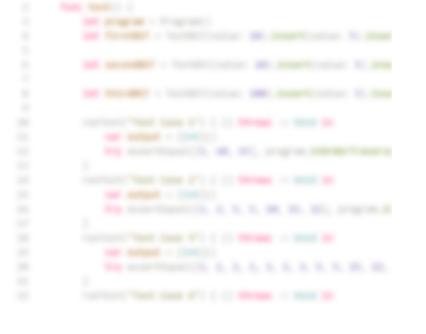
```
Run Code
```

Your Solutions

Run Code

```
Solution 1
 1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
 3
   class Program {
       class BST {
 5
           var value: Int?
           var left: BST?
6
 7
           var right: BST?
9
            init(value: Int) {
10
                self.value = value
                left = nil
11
12
                right = nil
13
14
15
16
       // O(n) time | O(n) space
17
       func inOrderTraversal(tree: BST?, array: inout [Int]) -> [In
            if tree !== nil {
18
19
                inOrderTraversal(tree: tree?.left, array: &array)
20
                if let value = tree?.value {
21
                    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]) -> [I
33
            if tree !== nil {
```

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



Run or submit code when you're ready.