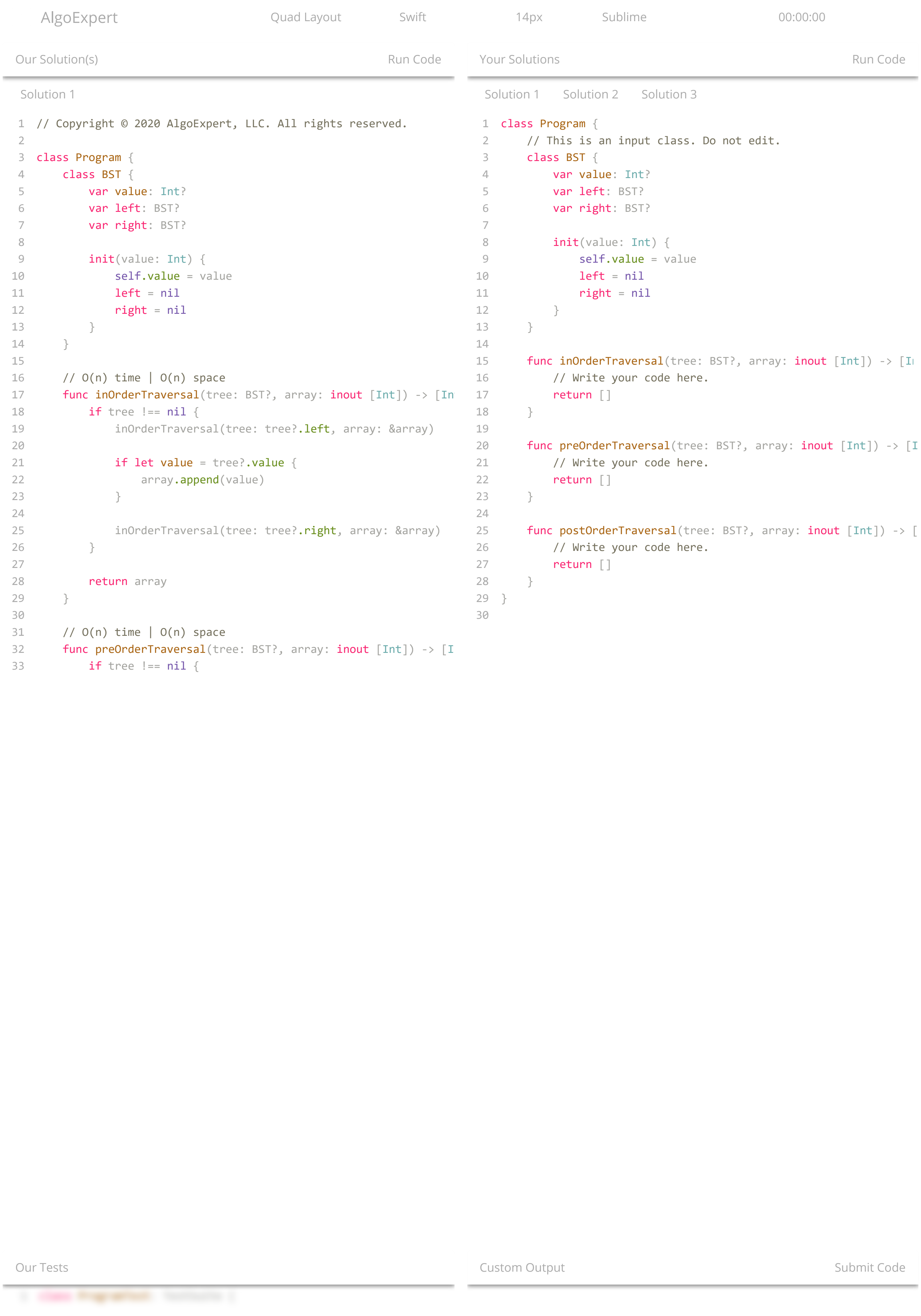

```swift
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

class Program {
    class BST {
        var value: Int?
        var left: BST?
        var right: BST?

        init(value: Int) {
            self.value = value
            left = nil
            right = nil
        }
    }

    // O(n) time | O(n) space
    func inOrderTraversal(tree: BST?, array: inout [Int]) -> [In
        if tree !== nil {
            inOrderTraversal(tree: tree?.left, array: &array)

            if let value = tree?.value {
                array.append(value)
            }

            inOrderTraversal(tree: tree?.right, array: &array)
        }

        return array
    }

    // O(n) time | O(n) space
    func preOrderTraversal(tree: BST?, array: inout [Int]) -> [I
        if tree !== nil {
```

```swift
class Program {
    // This is an input class. Do not edit.
    class BST {
        var value: Int?
        var left: BST?
        var right: BST?

        init(value: Int) {
            self.value = value
            left = nil
            right = nil
        }
    }

    func inOrderTraversal(tree: BST?, array: inout [Int]) -> [I
        // Write your code here.
        return []
    }

    func preOrderTraversal(tree: BST?, array: inout [Int]) -> [I
        // Write your code here.
        return []
    }

    func postOrderTraversal(tree: BST?, array: inout [Int]) -> [
        // Write your code here.
        return []
    }
}
```

Run or submit code when you're ready.