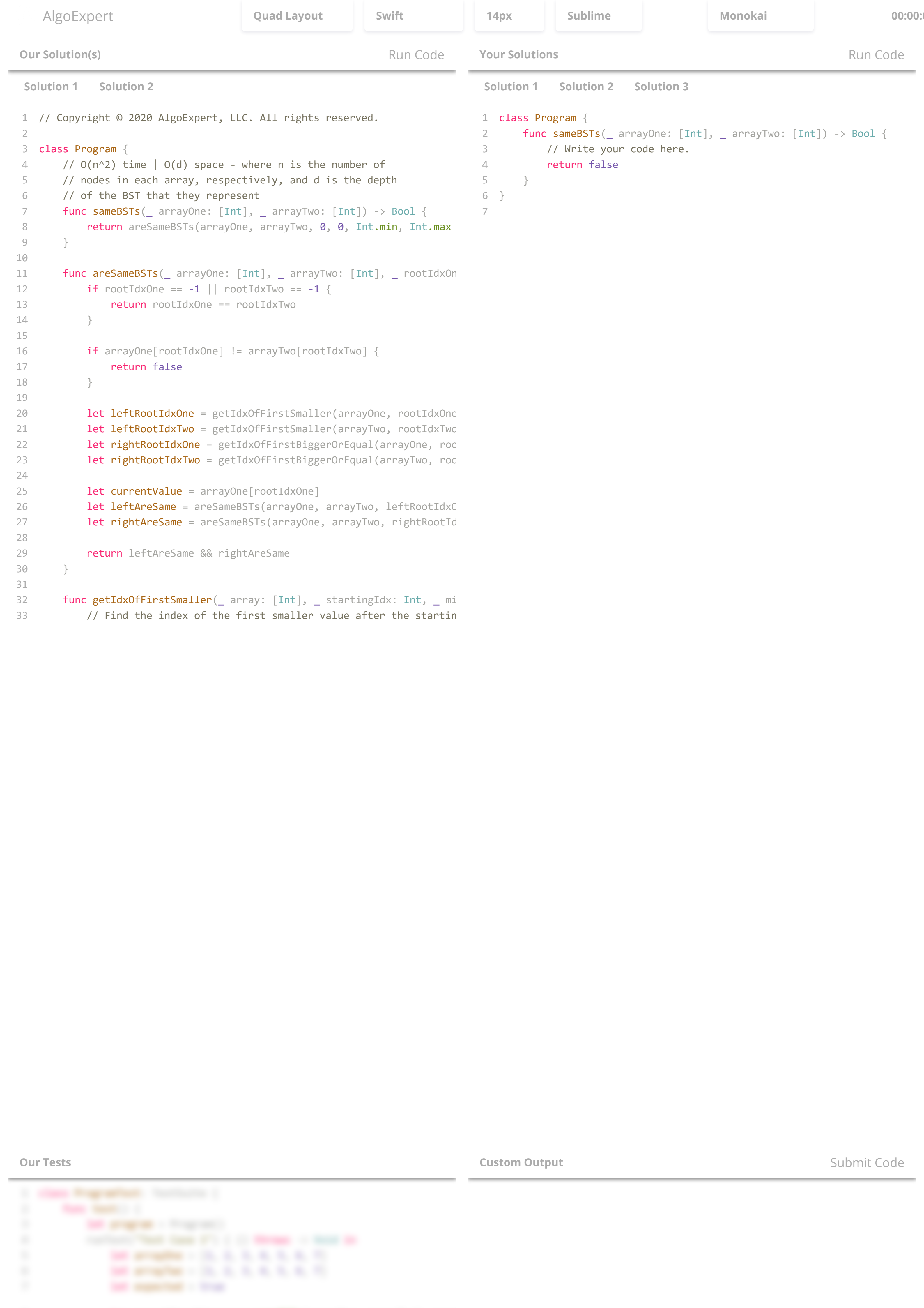

```swift
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

class Program {
    // O(n^2) time | O(d) space - where n is the number of
    // nodes in each array, respectively, and d is the depth
    // of the BST that they represent
    func sameBSTs(_ arrayOne: [Int], _ arrayTwo: [Int]) -> Bool {
        return areSameBSTs(arrayOne, arrayTwo, 0, 0, Int.min, Int.max
    }

    func areSameBSTs(_ arrayOne: [Int], _ arrayTwo: [Int], _ rootIdxOn
        if rootIdxOne == -1 || rootIdxTwo == -1 {
            return rootIdxOne == rootIdxTwo
        }

        if arrayOne[rootIdxOne] != arrayTwo[rootIdxTwo] {
            return false
        }

        let leftRootIdxOne = getIdxOfFirstSmaller(arrayOne, rootIdxOne
        let leftRootIdxTwo = getIdxOfFirstSmaller(arrayTwo, rootIdxTwo
        let rightRootIdxOne = getIdxOfFirstBiggerOrEqual(arrayOne, roc
        let rightRootIdxTwo = getIdxOfFirstBiggerOrEqual(arrayTwo, roc

        let currentValue = arrayOne[rootIdxOne]
        let leftAreSame = areSameBSTs(arrayOne, arrayTwo, leftRootIdxC
        let rightAreSame = areSameBSTs(arrayOne, arrayTwo, rightRootId

        return leftAreSame && rightAreSame
    }

    func getIdxOfFirstSmaller(_ array: [Int], _ startingIdx: Int, _ mi
        // Find the index of the first smaller value after the startin
```

```swift
class Program {
    func sameBSTs(_ arrayOne: [Int], _ arrayTwo: [Int]) -> Bool {
        // Write your code here.
        return false
    }
}
```

Run or submit code when you're ready.