Our Solution(s) Run Code

Solution 1 Solution 2

```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
        }
    }

    // Average: O(log(n)) time | O(1) space
    // Worst: O(n) time | O(1) space
    func findClosestValueInBST(tree: BST?, target: Int) -> Int {
        var closest = Int(Int32.max)

        return findClosestValueInBSTHelper(tree: tree, target: target, closest: &
    }

    func findClosestValueInBSTHelper(tree: BST?, target: Int,
                                     closest: inout Int) -> Int {
        var currentNode = tree

        while currentNode !== nil {
            if let node = currentNode {
                let closestDifference = target - closest
                let currentDifference = target - node.value

                if closestDifference.magnitude > currentDifference.magnitude {
                    closest = node.value
                }
            }

            if let node = currentNode, target < node.value {
                currentNode = node.left
            } else if let node = currentNode, target > node.value {
                currentNode = node.right
            } else {
                break
            }
        }

        return closest
    }
}

```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

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

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

    func findClosestValueInBST(tree: BST?, target: Int) -> Int {
        // Write your code here.
        return -1
    }
}

```

Custom Output Raw Output Submit Code

Run or submit code when you're ready.