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

class Program {
    // O(log(n)) time | O(log(n)) space
    func searchForRange(_ array: [Int], _ target: Int) -> [Int] {
        var finalRange = [-1, -1]

        alteredBinarySearch(array, target, 0, array.count - 1, &finalRange, true)
        alteredBinarySearch(array, target, 0, array.count - 1, &finalRange, false)
        return finalRange
    }

    func alteredBinarySearch(_ array: [Int], _ target: Int, _ leftPointer: Int, _ rightPointer: Int, _ finalRange: inout [Int], _ goLeft: Bool) {
        if leftPointer > rightPointer {
            return
        }

        let middle = (leftPointer + rightPointer) / 2

        if array[middle] > target {
            alteredBinarySearch(array, target, leftPointer, middle - 1, &finalRange, goLeft)
        } else if array[middle] < target {
            alteredBinarySearch(array, target, middle + 1, rightPointer, &finalRange, goLeft)
        } else {
            if goLeft {
                if middle == 0 || array[middle] != array[middle - 1] {
                    finalRange[0] = middle
                } else {
                    alteredBinarySearch(array, target, leftPointer, middle - 1, &finalRange, goLeft)
                }
            } else {
                if middle == array.count - 1 || array[middle] != array[middle + 1] {
                    finalRange[1] = middle
                } else {
                    alteredBinarySearch(array, target, middle + 1, rightPointer, &finalRange, goLeft)
                }
            }
        }
    }
}
```