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

class Program {
    // O(log(n)) time | O(1) space
    func shiftedBinarySearch(_ array: [Int], _ target: Int) -> Int {
        var leftPointer = 0
        var rightPointer = array.count - 1

        return shiftedBinarySearchHelper(array, target, &leftPointer, &rightPointer)
    }

    func shiftedBinarySearchHelper(_ array: [Int], _ target: Int, _ leftPointer: inout Int, _ rightPointer: inout Int) -> Int {
        while leftPointer <= rightPointer {
            let middle = (leftPointer + rightPointer) / 2
            let potentialMatch = array[middle]
            let leftNumber = array[leftPointer]
            let rightNumber = array[rightPointer]

            if target == potentialMatch {
                return middle
            } else if leftNumber < potentialMatch {
                if target < potentialMatch, target >= leftNumber {
                    rightPointer = middle - 1
                } else {
                    leftPointer = middle + 1
                }
            } else {
                if target <= rightNumber, target > potentialMatch {
                    leftPointer = middle + 1
                } else {
                    rightPointer = middle - 1
                }
            }
        }

        return -1
    }
}

```