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

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

    func shiftedBinarySearchHelper(_ array: [Int], _ target: Int, _ leftPointer: Int, _ rightPointer: Int) -> Int {
        if leftPointer > rightPointer {
            return -1
        }

        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 {
                return shiftedBinarySearchHelper(array, target, leftPointer, middle - 1)
            } else {
                return shiftedBinarySearchHelper(array, target, middle + 1, rightPointer)
            }
        } else {
            if target <= rightNumber, target > potentialMatch {
                return shiftedBinarySearchHelper(array, target, middle + 1, rightPointer)
            } else {
                return shiftedBinarySearchHelper(array, target, leftPointer, middle - 1)
            }
        }
    }
}

```