Our Solution(s)

```
Run Code
```

Your Solutions

Run Code

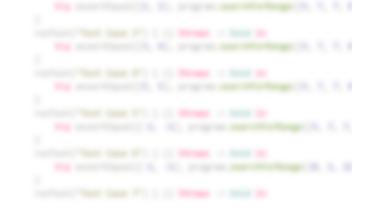
```
Solution 1 Solution 2
 1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
   class Program {
       // O(\log(n)) time | O(\log(n)) space
4
        func searchForRange(_ array: [Int], _ target: Int) -> [Int] {
            var finalRange = [-1, -1]
            alteredBinarySearch(array, target, 0, array.count - 1, &final
9
            alteredBinarySearch(array, target, 0, array.count - 1, &final
10
            return finalRange
11
12
13
        func alteredBinarySearch(_ array: [Int], _ target: Int, _ leftPoi
14
            if leftPointer > rightPointer {
16
17
           let middle = (leftPointer + rightPointer) / 2
18
19
20
            if array[middle] > target {
21
                alteredBinarySearch(array, target, leftPointer, middle - {\tt 1}
            } else if array[middle] < target {</pre>
               alteredBinarySearch(array, target, middle + 1, rightPointe
                if goLeft {
26
                    if middle == 0 || array[middle] != array[middle - 1] {
27
                        finalRange[0] = middle
28
29
                        alteredBinarySearch(array, target, leftPointer, mi
30
31
                } else {
                    if middle == array.count - 1 || array[middle] != array
33
                        finalRange[1] = middle
```

```
class Program {
  func searchForRange(_ array: [Int], _ target: Int) -> [Int] {
     // Write your code here.
     return []
}
}
```

Our Tests

Custom Output

Submit Code



Run or submit code when you're ready.