AlgoExpert Quad Layout Swift 12px Sublime Monokai 00:00:00

 Prompt
 Scratchpad
 Our Solution(s)
 Video Explanation

Run Code

```
_{\rm 1} \, // Copyright 0 2020 AlgoExpert, LLC. All rights reserved.
 3 class Program {
        // O(\log(n)) time | O(1) space
        func searchForRange(_ array: [Int], _ target: Int) -> [Int] {
            var finalRange = [-1, -1]
            var leftPointer = 0
9
            var rightPointer = array.count - 1
10
11
            {\tt alteredBinarySearch(array,\ target,\ \&leftPointer,\ \&rightPointer,\ \&finalRange,\ true)}
12
13
            leftPointer = 0
14
            rightPointer = array.count - 1
15
            {\tt alteredBinarySearch(array,\ target,\ \&leftPointer,\ \&rightPointer,\ \&finalRange,\ false)}
16
            return finalRange
17
18
19
        func alteredBinarySearch(_ array: [Int], _ target: Int, _ leftPointer: inout Int, _ rightPointer: inout Int, _ finalRange: inout [Int], _ goLeft: Bool) {
20
            while leftPointer <= rightPointer {</pre>
21
                let middle = (leftPointer + rightPointer) / 2
22
23
                if array[middle] > target {
24
                    rightPointer = middle - 1
25
                } else if array[middle] < target {</pre>
26
                    leftPointer = middle + 1
27
                } else {
                    if goLeft {
28
                        if middle == 0 || array[middle] != array[middle - 1] {
29
30
                            finalRange[0] = middle
31
                            return
32
                        } else {
33
                            rightPointer = middle - 1
                    } else {
35
                        if middle == array.count - 1 || array[middle] != array[middle + 1] {
36
37
                            finalRange[1] = middle
38
                            return
39
                        } else {
40
                            leftPointer = middle + 1
42
                   }
43
44
           }
45
```

Solution 1 Solution 2