

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class Program {
4     // O(nlogn) time | O(n) space
5     func longestIncreasingSubsequence(_ array: [Int]) -> [Int] {
6         var length = 0
7         var indices: [Int?] = Array(repeating: nil, count: array.count)
8         var sequences: [Int?] = Array(repeating: nil, count: array.count)
9
10        for i in 0 ..< array.count {
11            var startIndex = 1
12            var endIndex = length
13            let number = array[i]
14            let newLength = binarySearch(&startIndex, &endIndex, indices, number)
15
16            indices[newLength] = i
17            sequences[i] = indices[newLength - 1]
18            length = max(length, newLength)
19        }
20
21        return buildSequence(array, sequences, &indices[length])
22    }
23
24    func binarySearch(_ startIndex: inout Int, _ endIndex: inout Int,
25                     indices: [Int?], number: Int) -> Int {
26        if startIndex > endIndex {
27            return startIndex
28        }
29
30        var middleIndex = Double(startIndex + endIndex) / 2
31        middleIndex = middleIndex.rounded(.down)
32        let intMiddle = Int(middleIndex)
33
34        if let index = indices[intMiddle] {
35            if number > array[index] {
36                startIndex = intMiddle + 1
37            } else {
38                endIndex = intMiddle
39            }
40        }
41
42        return binarySearch(&startIndex, &endIndex, indices, number)
43    }
44
45    func buildSequence(array: [Int], indices: [Int?], start: Int) -> [Int] {
46        var sequence: [Int] = []
47        var index = start
48        while index < array.count {
49            sequence.append(array[index])
50            if let nextIndex = indices[index] {
51                index = nextIndex
52            } else {
53                break
54            }
55        }
56        return sequence
57    }
58 }
```

Solution 1

Solution 2

Solution 3

```
1 class Program {
2     func longestIncreasingSubsequence(_ array: [Int]) -> [Int] {
3         // Write your code here.
4         return []
5     }
6 }
7
```

```

18         self.event_handlers[id].append(handler)
19     }
20     return "Next time I'll do it" if id is None else "Next"
21     def event_handler(id, h, id), arg in self.event_handlers[id]:
22     }
23     return "Next time I'll do it" if id is None else "Next"
24     def event_handler(id, h, id), arg in self.event_handlers[id]:
25     }
26     return "Next time I'll do it" if id is None else "Next"
27     def event_handler(id, h, id), arg in self.event_handlers[id]:
28     }
29     return "Next time I'll do it" if id is None else "Next"
30     def event_handler(id, h), arg in self.event_handlers[id]:
31     }
32     return "Next time I'll do it" if id is None else "Next"

```

Run or submit code when you're ready.