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

Prompt Scratchpad Our Solution(s) Video Explanation Run Code

```
Solution 1 Solution 2
```

```
_{\rm 1} \, // Copyright 0 2020 AlgoExpert, LLC. All rights reserved.
  3 package main
  5 // O(n^2) time | O(n) space
  6 func LongestIncreasingSubsequence(array []int) []int {
       sequences := make([]int, len(array))
       lengths := make([]int, len(array))
       for i := range array {
         sequences[i] = -1
10
          lengths[i] = 1
11
12
13
       for i := range array {
         currentNum := array[i]
14
15
          for j := 0; j < i; j++ {
16
           otherNum := array[j]
17
            \textbf{if} \ \text{otherNum} \ < \ \text{currentNum} \ \&\& \ \text{lengths[j]+1} \ >= \ \text{lengths[i]} \ \{ \\
             lengths[i] = lengths[j] + 1
18
              sequences[i] = j
19
20
21
22
         }
23
       maxLengthIndex := 0
24
       for i := range array {
25
         if lengths[i] > lengths[maxLengthIndex] {
26
           maxLengthIndex = i
27
28
29
       return buildSequence(array, sequences, maxLengthIndex)
30 }
31
     func buildSequence(array, sequences []int, index int) []int {
33
       out := []int{}
34
       for index != -1 {
         out = append(out, array[index])
35
36
         index = sequences[index]
37
38
       reverse(out)
39
       return out
40 }
41
func reverse(numbers []int) {
for i, j := 0, len(numbers)-1; i < j; i, j = i+1, j-1 {
    numbers[i], numbers[j] = numbers[j], numbers[i]
}</pre>
45
46 }
```