

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1	Solution 2	Solution 1	Solution 2	Solution 3
<pre>1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved. 2 3 class Program { 4     // O(n^2) time   O(n) space 5     func longestIncreasingSubsequence(_ array: [Int]) -&gt; [Int] { 6         var indexOfMaxLength: Int? = 0 7         var lengths = Array(repeating: 1, count: array.count) 8         var sequences: [Int?] = Array(repeating: nil, count: array.count) 9 10        for i in 0 ..&lt; array.count { 11            let currentNumber = array[i] 12 13            for j in 0 ..&lt; i { 14                let otherNumber = array[j] 15 16                if otherNumber &lt; currentNumber, lengths[i] &lt;= lengths[j] { 17                    lengths[i] = lengths[j] + 1 18                    sequences[i] = j 19                } 20            } 21 22            if lengths[i] &gt; lengths[indexOfMaxLength!] { 23                indexOfMaxLength = i 24            } 25        } 26 27        return buildSequence(array, sequences, &amp;indexOfMaxLength) 28    } 29 30    func buildSequence(_ array: [Int], _ sequences: [Int?], _ currentIndex: Int) -&gt; [Int] { 31        var sequence = [Int]() 32 33        while currentIndex != nil { 34            sequence.append(array[currentIndex!]) 35            currentIndex = sequences[currentIndex!]</pre>		<pre>1 class Program { 2     func longestIncreasingSubsequence(_ array: [Int]) -&gt; [Int] { 3         // Write your code here. 4         return [] 5     } 6 } 7</pre>		

Our Tests

Custom Output

Submit Code

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

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

```

18         self.eventQueue.put(self.program.getAgentForLearningState(
19
20             self.testState["state"] + self.obspace + self.act
21             self.eventQueue.put(self.A, self.B, self.obspace.getAgentForLearningState(
22
23             self.testState["state"] + self.obspace + self.act
24             self.eventQueue.put(self.A, self.B, self.obspace.getAgentForLearningState(
25
26             self.testState["state"] + self.obspace + self.act
27             self.eventQueue.put(self.A, self.B, self.obspace.getAgentForLearningState(
28
29             self.testState["state"] + self.obspace + self.act
30             self.eventQueue.put(self.A, self.obspace.getAgentForLearningState(
31
32             self.testState["state"] + self.obspace + self.act

```

Run or submit code when you're ready.