

Solution 1Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 using System;
4 using System.Collections.Generic;
5
6 public class Program {
7     // O(n^2) time | O(n) space
8     public static List<int> LongestIncreasingSubsequence(int[] array) {
9         int[] sequences = new int[array.Length];
10        Array.Fill(sequences, Int32.MinValue);
11        int[] lengths = new int[array.Length];
12        Array.Fill(lengths, 1);
13        int maxLengthIdx = 0;
14        for (int i = 0; i < array.Length; i++) {
15            int currentNum = array[i];
16            for (int j = 0; j < i; j++) {
17                int otherNum = array[j];
18                if (otherNum < currentNum && lengths[j] + 1 >= lengths[i]) {
19                    lengths[i] = lengths[j] + 1;
20                    sequences[i] = j;
21                }
22            }
23            if (lengths[i] >= lengths[maxLengthIdx]) {
24                maxLengthIdx = i;
25            }
26        }
27        return buildSequence(array, sequences, maxLengthIdx);
28    }
29
30    public static List<int> buildSequence(int[] array, int[] sequences, int currentIdx) {
31        List<int> sequence = new List<int>();
32        while (currentIdx != Int32.MinValue) {
33            sequence.Insert(0, array[currentIdx]);
34            currentIdx = sequences[currentIdx];
35        }
36        return sequence;
37    }
38 }
39
```

