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

Prompt    Scratchpad    Our Solution(s)    Video Explanation    Run Code

Solution 1    Solution 2

```java
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

import java.util.*;

class Program {
  // O(n^2) time | O(n) space
  public static List<Integer> longestIncreasingSubsequence(int[] array) {
    int[] sequences = new int[array.length];
    Arrays.fill(sequences, Integer.MIN_VALUE);
    int[] lengths = new int[array.length];
    Arrays.fill(lengths, 1);
    int maxLengthIdx = 0;
    for (int i = 0; i < array.length; i++) {
      int currentNum = array[i];
      for (int j = 0; j < i; j++) {
        int otherNum = array[j];
        if (otherNum < currentNum && lengths[j] + 1 >= lengths[i]) {
          lengths[i] = lengths[j] + 1;
          sequences[i] = j;
        }
      }
      if (lengths[i] >= lengths[maxLengthIdx]) {
        maxLengthIdx = i;
      }
    }
    return buildSequence(array, sequences, maxLengthIdx);
  }

  public static List<Integer> buildSequence(int[] array, int[] sequences, int currentIdx) {
    List<Integer> sequence = new ArrayList<Integer>();
    while (currentIdx != Integer.MIN_VALUE) {
      sequence.add(0, array[currentIdx]);
      currentIdx = sequences[currentIdx];
    }
    return sequence;
  }
}
```