AlgoExpert

Quad Layout | Python | 12px | Sublime | Monokai | 00:00:00

Prompt | Scratchpad | Our Solution(s) | Video Explanation | Run Code

Solution 1 | Solution 2

```python
# Copyright © 2020 AlgoExpert, LLC. All rights reserved.

# O(n^2) time | O(n) space
def longestIncreasingSubsequence(array):
    sequences = [None for x in array]
    lengths = [1 for x in array]
    maxLengthIdx = 0
    for i in range(len(array)):
        currentNum = array[i]
        for j in range(0, i):
            otherNum = array[j]
            if otherNum < currentNum and 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)


def buildSequence(array, sequences, currentIdx):
    sequence = []
    while currentIdx is not None:
        sequence.append(array[currentIdx])
        currentIdx = sequences[currentIdx]
    return list(reversed(sequence))

```