AlgoExpert

Quad Layout | C++ | 12px | Sublime | Monokai | 00:00:00

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

Solution 1 | Solution 2

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

#include <vector>
#include <climits>
using namespace std;

vector<int> buildSequence(vector<int> array, vector<int> sequences,
                          int currentIdx);

// O(n^2) time | O(n) space
vector<int> longestIncreasingSubsequence(vector<int> array) {
  vector<int> sequences(array.size(), INT_MIN);
  vector<int> lengths(array.size(), 1);
  int maxLengthIdx = 0;
  for (int i = 0; i < array.size(); 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);
}

vector<int> buildSequence(vector<int> array, vector<int> sequences,
                          int currentIdx) {
  vector<int> sequence;
  while (currentIdx != INT_MIN) {
    sequence.insert(sequence.begin(), array[currentIdx]);
    currentIdx = sequences[currentIdx];
  }
  return sequence;
}
```