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

Prompt Scratchpad Our Solution(s) Video Explanation Run Code

Solution 1

41

```
1\, // Copyright @ 2020 AlgoExpert, LLC. All rights reserved.
   import java.util.*;
 5 class Program {
     // O(n^2) time | O(n) space
      public static List<List<Integer>> maxSumIncreasingSubsequence(int[] array) {
        int[] sequences = new int[array.length];
        Arrays.fill(sequences, Integer.MIN_VALUE);
10
        int[] sums = array.clone();
        int maxSumIdx = 0;
11
12
        for (int i = 0; i < array.length; i++) {</pre>
          int currentNum = array[i];
13
14
          for (int j = 0; j < i; j++) {
            int otherNum = array[j];
15
16
             \textbf{if} \ (\texttt{otherNum} \ < \ \texttt{currentNum} \ \&\& \ \texttt{sums[j]} \ + \ \texttt{currentNum} \ >= \ \texttt{sums[i]}) \ \{ \\
17
              sums[i] = sums[j] + currentNum;
              sequences[i] = j;
18
19
20
21
          if (sums[i] >= sums[maxSumIdx]) {
22
            maxSumIdx = i;
23
24
25
        return buildSequence(array, sequences, maxSumIdx, sums[maxSumIdx]);
26
27
      public static List<List<Integer>> buildSequence(
28
29
          int[] array, int[] sequences, int currentIdx, int sums) {
30
        List<List<Integer>> sequence = new ArrayList<List<Integer>>();
31
        sequence.add(new ArrayList<Integer>());
        sequence.add(new ArrayList<Integer>());
32
        sequence.get(0).add(sums);
33
        while (currentIdx != Integer.MIN_VALUE) {
34
35
          sequence.get(1).add(0, array[currentIdx]);
36
          currentIdx = sequences[currentIdx];
37
        return sequence;
38
39
40 }
```