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

 Prompt
 Scratchpad
 Our Solution(s)
 Video Explanation

Run Code

```
_{\rm 1} \, // Copyright 0 2020 AlgoExpert, LLC. All rights reserved.
    3 class Program {
                         // O(2^{n} + m)) time | O(n + m) space - where n is the length
                          \ensuremath{//} of the first string and \ensuremath{\mathrm{m}} is the length of the second string
                         func interweavingStrings(_ one: String, _ two: String, _ three: String) -> Bool {
   if three.length != one.length + two.length {
                                                return false
  9
                                       return areInterwoven(one, two, three, 0, 0)
 10
11
12
13
                          \label{function} \mbox{func areInterwoven}(\cline{Limits_1}\cline{Limits_2}\cline{Limits_3}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\cline{Limits_4}\
14
                                      let k = i + j
                                      if k == three.length {
15
16
                                               return true
17
18
19
                                      let oneI = one.index(one.startIndex, offsetBy: i)
20
                                      let twoJ = two.index(two.startIndex, offsetBy: j)
21
                                      let threeK = three.index(three.startIndex, offsetBy: k)
22
23
                                      if i < one.length, one[oneI] == three[threeK] {</pre>
                                                  if areInterwoven(one, two, three, i + 1, j) {
24
                                                              return true
25
26
27
28
                                     if j < two.length, two[twoJ] == three[threeK] {</pre>
29
30
31
                                                  32
                                      return false
33
```

Solution 1 Solution 2

35