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

Prompt Scratchpad Our Solution(s) Video Explanation Run Code

Solution 1 Solution 2

48 }

```
1\, // Copyright @ 2020 AlgoExpert, LLC. All rights reserved.
   using namespace std;
 5 bool areInterwoven(string one, string two, string three, int i, int j,
                      vector<vector<int>> &cache);
 8 // O(nm) time | O(nm) space - where n is the length of the
9\, // first string and m is the length of the second string
10 bool interweavingStrings(string one, string two, string three) {
    if (three.size() != one.size() + two.size()) {
11
12
       return false;
13
14
15
      vector<vector<int>> cache;
      for (int i = 0; i < one.size() + 1; i++) {</pre>
16
17
       cache.push_back(vector<int>{});
        for (int j = 0; j < two.size() + 1; j++) {</pre>
18
19
         cache[i].push_back(-1);
20
21
22
23
      return areInterwoven(one, two, three, 0, 0, cache);
24 }
25
26 bool areInterwoven(string one, string two, string three, int i, int j,
                      vector<vector<int>> &cache) {
27
28
      if (cache[i][j] != -1)
       return cache[i][j];
29
30
31
      int k = i + j;
     if (k == three.size())
32
       return true;
33
     if (i < one.size() && one[i] == three[k]) {</pre>
35
       cache[i][j] = areInterwoven(one, two, three, i + 1, j, cache);
36
37
        if (cache[i][j] == true)
38
         return true;
39
40
      if (j < two.size() && two[j] == three[k]) {</pre>
41
       cache[i][j] = areInterwoven(one, two, three, i, j + 1, cache);
43
       return cache[i][j];
44
45
     cache[i][j] = false;
46
47
     return false;
```