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

Prompt Scratchpad Our Solution(s) Video Explanation Run Code

Solution 1 Solution 2 Solution 3 Solution 4

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

#include <vector>
using namespace std;

vector<char> buildSequence(vector<vector<vector<int>>> lcs);

// O(nm) time | O(nm) space
vector<char> longestCommonSubsequence(string str1, string str2) {
  vector<vector<vector<int>>> lcs(
      str2.length() + 1,
      vector<vector<int>>(str1.length() + 1, vector<int>(4, 0)));
  for (int i = 1; i < str2.length() + 1; i++) {
    for (int j = 1; j < str1.length() + 1; j++) {
      if (str2[i - 1] == str1[j - 1]) {
        lcs[i][j] = {str2[i - 1], lcs[i - 1][j - 1][1] + 1, i - 1, j - 1};
      } else {
        if (lcs[i - 1][j][1] > lcs[i][j - 1][1]) {
          lcs[i][j] = {-1, lcs[i - 1][j][1], i - 1, j};
        } else {
          lcs[i][j] = {-1, lcs[i][j - 1][1], i, j - 1};
        }
      }
    }
  }
  return buildSequence(lcs);
}

vector<char> buildSequence(vector<vector<vector<int>>> lcs) {
  vector<char> sequence;
  int i = lcs.size() - 1;
  int j = lcs[0].size() - 1;
  while (i != 0 && j != 0) {
    vector<int> currentEntry = lcs[i][j];
    if (currentEntry[0] != -1) {
      sequence.insert(sequence.begin(), currentEntry[0]);
    }
    i = currentEntry[2];
    j = currentEntry[3];
  }
  return sequence;
}
```