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

Prompt Scratchpad Our Solution(s) Video Explanation Run Code

Solution 1 Solution 2 Solution 3 Solution 4

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

// O(nm) time | O(nm) space
function longestCommonSubsequence(str1, str2) {
  const lcs = [];
  for (let i = 0; i < str2.length + 1; i++) {
    const row = [];
    for (let j = 0; j < str1.length + 1; j++) {
      const entry = new Array(4);
      entry[1] = 0;
      row.push(entry);
    }
    lcs.push(row);
  }
  for (let i = 1; i < str2.length + 1; i++) {
    for (let 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] = [null, lcs[i - 1][j][1], i - 1, j];
        } else {
          lcs[i][j] = [null, lcs[i][j - 1][1], i, j - 1];
        }
      }
    }
  }
  return buildSequence(lcs);
}

function buildSequence(lcs) {
  const sequence = [];
  let i = lcs.length - 1;
  let j = lcs[0].length - 1;
  while (i !== 0 && j !== 0) {
    let currentEntry = lcs[i][j];
    if (currentEntry[0]) {
      sequence.unshift(currentEntry[0]);
    }
    i = currentEntry[2];
    j = currentEntry[3];
  }
  return sequence;
}

exports.longestCommonSubsequence = longestCommonSubsequence;
```