

PromptScratchpadOur Solution(s)Video Explanation

Run Code

Solution 1Solution 2Solution 3Solution 4

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 // O(nm*min(n, m)) time | O((min(n, m))^2) space
4 function longestCommonSubsequence(str1, str2) {
5   const small = str1.length < str2.length ? str1 : str2;
6   const big = str1.length >= str2.length ? str1 : str2;
7   const evenLcs = new Array(small.length + 1).fill([]);
8   const oddLcs = new Array(small.length + 1).fill([]);
9   for (let i = 1; i < big.length + 1; i++) {
10     let currentLcs, previousLcs;
11     if (i % 2 === 1) {
12       currentLcs = oddLcs;
13       previousLcs = evenLcs;
14     } else {
15       currentLcs = evenLcs;
16       previousLcs = oddLcs;
17     }
18     for (let j = 1; j < small.length + 1; j++) {
19       if (big[i - 1] === small[j - 1]) {
20         currentLcs[j] = previousLcs[j - 1].concat(big[i - 1]);
21       } else {
22         currentLcs[j] = previousLcs[j].length > currentLcs[j - 1].length ? previousLcs[j] : currentLcs[j - 1];
23       }
24     }
25   }
26   return big.length % 2 === 0 ? evenLcs[small.length] : oddLcs[small.length];
27 }
28
29 exports.longestCommonSubsequence = longestCommonSubsequence;
30
```