Solution 1   Solution 2   Solution 3   Solution 4

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