

## Longest Common Subsequence

Difficulty: **Medium**Accuracy: **41.68%**Submissions: **306K+**Points: **4**

Given two strings **s1** and **s2**, return the length of their **longest common subsequence** (LCS). If there is no common subsequence, return **0**.

*A subsequence is a sequence that can be derived from the given string by deleting some or no elements without changing the order of the remaining elements. For example, "ABE" is a subsequence of "ABCDE".*

### Examples:

**Input:** s1 = "ABCDGH", s2 = "AEDFHR"

**Output:** 3

**Explanation:** The longest common subsequence of "ABCDGH" and "AEDFHR" is "ADH", which has a length of 3.

**Input:** s1 = "ABC", s2 = "AC"

**Output:** 2

**Explanation:** The longest common subsequence of "ABC" and "AC" is "AC", which has a length of 2.

```
1 // } Driver Code Ends
27 class Solution {
28     static int lcs(String s1, String s2) {
29         int m = s1.length();
30         int n = s2.length();
31         int[][] dp = new int[m+1][n+1];
32         for(int i = 1; i<=m ; i++){
33             for(int j = 1; j<=n ;j++){
34                 if(s1.charAt(i-1) == s2.charAt(j-1)){
35                     dp[i][j] = dp[i-1][j-1] + 1;
36                 }
37                 else{
38                     dp[i][j] = Math.max(dp[i-1][j],dp[i][j-1]);
39                 }
40             }
41         }
42         return dp[m][n];
43     }
44 }
```

