3-DP-Longest Common Subsequence

Aim:

Given two strings find the length of the common longest subsequence(need not be contiguous) between the two.

Example:

 s1: ggtabe

 s2: tgatasb

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| s1 |  | a | g | **g** | **t** | **a** | **b** |  |
| s2 |  | **g** | x | **t** | x | **a** | y | **b** |

**The length is 4**

Solveing it using Dynamic Programming

**For example:**

| **Input** | **Result** |
| --- | --- |
| aab  azb | 2 |

Answer:(penalty regime: 0 %)

Algorithm:

1. Read the strings s1 and s2.
2. Initialize a DP table dp of size (m+1) x (n+1) where m and n are the lengths of s1 and s2, respectively.
3. Fill the DP table using dynamic programming: if characters match, increment the value from the diagonal; otherwise, take the maximum from the top or left cell.
4. The value at dp[m][n] is the length of the longest common subsequence.
5. Print the result.

Code:

#include <stdio.h>

#include <string.h>

int longestCommonSubsequence(char s1[], char s2[]) {

int m = strlen(s1);

int n = strlen(s2);

int dp[m + 1][n + 1];

// Initialize the DP table with base cases

for (int i = 0; i <= m; i++) {

for (int j = 0; j <= n; j++) {

if (i == 0 || j == 0) {

dp[i][j] = 0;

}

else if (s1[i - 1] == s2[j - 1]) {

dp[i][j] = dp[i - 1][j - 1] + 1;

}

else {

dp[i][j] = (dp[i - 1][j] > dp[i][j - 1]) ? dp[i - 1][j] : dp[i][j - 1];

}

}

}

return dp[m][n];

}

int main() {

char s1[100], s2[100];

scanf("%s", s1);

scanf("%s", s2);

int result = longestCommonSubsequence(s1, s2);

printf("%d", result);

}

Output:

|  | **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- | --- |
|  | aab  azb | 2 | 2 |  |
|  | ABCD  ABCD | 4 | 4 |  |

Passed all tests!

**Correct**

Marks for this submission: 1.00/1.00.

Result:

The expected output was obtained