

LONGEST COMMON SUBSEQUENCE

84

Question :

Given two strings text1 and text2,
return the length of their longest
common subsequence.

If there is no common subsequence
return 0.

A subsequence of a string is a new
string generated from the original
string with some characters (can be
none) deleted w/o changing the
relative order of the remaining
characters.

For ex: "ace" is a subsequence of
"abcde".

A common subsequence of two strings is a subsequence that is common to both strings.

Input: text1 = "abcde", text2 = "ace"

Output: 3

Explanation: The longest common subsequence is "ace" and its length is 3.

Constraints: $1 \leq \text{text1 length}, \text{text2 length} \leq 1000$

CODE

```

STRING S1, S2;
INT DP[1005][1005];

INT LCS (INT i, INT j)
{
    IF (i < 0 || j < 0) RETURN 0;
    IF (DP[i][j] != -1) RETURN DP[i][j];
    → Remove 1 char from S1
    INT ANS = LCS (i-1, j);
    → Remove 1 char from S2
    ANS = MAX (ANS, LCS (i-1, j-1) + (S1[i] == S2[j]));
    RETURN DP[i][j] = ANS;
}

INT MAIN ()
{
    MEMSET (DP, -1, sizeof (DP));

```


s1 = "abcde";

s2 = "ace";

int ans = LCS (s1.size() - 1, s2.size() - 1);

cout << ans << "\n";

return 0;

}

→ ans = max (ans, LCS (i, j - 1));

→ Remove 1 char from s1 and s2