DAA LAB

Name: Atharva Jogiji
Section: A5_B2
Roll No: 34

PRACTICAL 5

Task-1

Code:

```
def LCS(X,Y):
  m=len(X)
  n=len(Y)
  dpTable=[[0]*(n+1) for _ in range(m+1)]
  for i in range(1,m+1):
      for j in range(1,n+1):
         if X[i-1]==Y[j-1]:
            dpTable[i][j]=dpTable[i-1][j-1]+1
         else:
            dpTable[i][j]=max(dpTable[i-1][j],dpTable[i][j-1])
  i,j=m,n
  LCS_string=[]
  while i>0 and j>0:
    if X[i-1]==Y[j-1]:
       LCS_string.append(X[i-1])
       i=i-1
       j=j-1
    elif dpTable[i-1][j]>dpTable[i][j - 1]:
       i=i-1
    else:
       j=j-1
  return dpTable, ''.join(reversed(LCS_string))

X=input("Enter 1st string: ")
Y=input("Enter 2nd string: ")
dpTable_table, LCS_string = LCS(X, Y)
print(" DP Matrix:")
for row in dpTable_table:
   print(row)
print("\nLongest Common Subsequence:", LCS_string)
print("Length of LCS:",len(LCS_string))
```

Output:

```
Enter 1st string: AGCCCTAAGGGCTACCTAGCTT
Enter 2nd string: GACAGCCTACAAGCGTTAGCTTG
 DP Matrix:
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
[0, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]
[0, 1, 1, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3]
[0, 1, 1, 2, 2, 2, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4]
[0, 1, 1, 2, 2, 2, 3, 4, 4, 4, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5]
[0, 1, 1, 2, 2, 2, 3, 4, 5, 5, 5, 5, 5, 5, 5, 5, 6, 6, 6, 6, 6, 6, 6, 6]
[0, 1, 2, 2, 3, 3, 3, 4, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 7, 7, 7, 7, 7, 7]
[0, 1, 2, 2, 3, 3, 3, 4, 5, 6, 6, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7]
[0, 1, 2, 2, 3, 4, 4, 4, 5, 6, 6, 7, 7, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8]
[0, 1, 2, 2, 3, 4, 4, 4, 5, 6, 6, 7, 7, 8, 8, 9, 9, 9, 9, 9, 9, 9, 9, 9]
[0, 1, 2, 2, 3, 4, 4, 4, 5, 6, 6, 7, 7, 8, 8, 9, 9, 9, 9, 10, 10, 10, 10, 10]
[0, 1, 2, 3, 3, 4, 5, 5, 5, 6, 7, 7, 7, 8, 9, 9, 9, 9, 9, 10, 11, 11, 11, 11]
[0, 1, 2, 3, 3, 4, 5, 5, 6, 6, 7, 7, 7, 8, 9, 9, 10, 10, 10, 10, 11, 12, 12, 12]
[0, 1, 2, 3, 4, 4, 5, 5, 6, 7, 7, 8, 8, 8, 9, 9, 10, 10, 11, 11, 11, 12, 12, 12]
[0, 1, 2, 3, 4, 4, 5, 6, 6, 7, 8, 8, 8, 8, 9, 9, 10, 10, 11, 11, 12, 12, 12, 12]
[0, 1, 2, 3, 4, 4, 5, 6, 6, 7, 8, 8, 8, 8, 9, 9, 10, 10, 11, 11, 12, 12, 12, 12]
[0, 1, 2, 3, 4, 4, 5, 6, 7, 7, 8, 8, 8, 8, 9, 9, 10, 11, 11, 11, 12, 13, 13, 13]
[0, 1, 2, 3, 4, 4, 5, 6, 7, 8, 8, 9, 9, 9, 9, 9, 10, 11, 12, 12, 12, 13, 13, 13]
[0, 1, 2, 3, 4, 5, 5, 6, 7, 8, 8, 9, 9, 10, 10, 10, 10, 11, 12, 13, 13, 13, 13, 14]
[0, 1, 2, 3, 4, 5, 6, 6, 7, 8, 9, 9, 9, 10, 11, 11, 11, 11, 12, 13, 14, 14, 14, 14]
[0, 1, 2, 3, 4, 5, 6, 6, 7, 8, 9, 9, 9, 10, 11, 11, 12, 12, 12, 13, 14, 15, 15, 15]
[0, 1, 2, 3, 4, 5, 6, 6, 7, 8, 9, 9, 9, 10, 11, 11, 12, 13, 13, 13, 14, 15, 16, 16]

Longest Common Subsequence: GCCCTAAGCTTAGCTT
Length of LCS: 16
```

TASK-2:

Code:

```
def LRS(a: str, b: str) -> str:
    n = len(a)
    m = len(b)

    c=[[0]*(m + 1) for _ in range(n+1)]
    for i in range(1, n+1):
        for j in range(1, m+1):
            if a[i-1] == b[j-1] and i!=j:
                c[i][j] = 1 + c[i-1][j-1]
            else:
                c[i][j] = max(c[i-1][j], c[i][j-1])

    print("DP Matrix:")
    print("    " + "  ".join(b))
    for i in range(n + 1):
        row = ["0" if i == 0 else a[i-1]]
        for j in range(m+1):
            row.append(str(c[i][j]))
        print("  ".join(row))

    i, j = n, m
```

```
    lrs = []
    while i > 0 and j > 0:
        if a[i-1] == b[j-1] and i!=j:
            lrs.append(a[i-1])
            i-=1
            j-=1
        elif c[i-1][j] >= c[i][j-1]:
            i-=1
        else:
            j-=1
    return "".join(reversed(lrs))

text = input("Enter string: ")
result = LRS(text, text)
print("Longest Repeated Subsequence:", result)
print("Length:", len(result))
```

Output:

```
Enter string: AABEBCDD
DP Matrix:
    A  A  B  E  B  C  D  D
0  0  0  0  0  0  0  0  0  0
A  0  0  1  1  1  1  1  1  1
A  0  1  1  1  1  1  1  1  1
B  0  1  1  1  1  2  2  2  2
E  0  1  1  1  1  2  2  2  2
B  0  1  1  2  2  2  2  2  2
C  0  1  1  2  2  2  2  2  2
D  0  1  1  2  2  2  2  2  3
D  0  1  1  2  2  2  2  3  3
Longest Repeated Subsequence: ABD
Length: 3
```

Start coding or generate with AI.

Leetcode:

Problem List

Submit

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 47 / 47 testcases passed

24jogijia submitted at Sep 16, 2025 16:42

Editorial | Solution

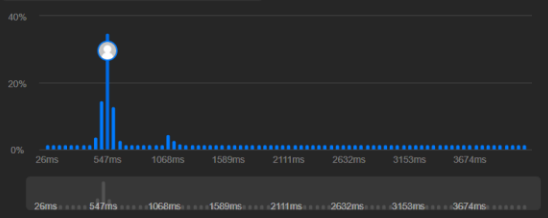
Runtime

548 ms | Beats 50.28%

Analyze Complexity

Memory

33.75 MB | Beats 53.22%



Code | Python

```python
class Solution(object):
    def longestCommonSubsequence(self, text1, text2):
        m=len(text1)
        n=len(text2)
        dp = [[0] * (n + 1) for _ in range(m + 1)]
```

Code

Python | Auto

```python
class Solution(object):
    def longestCommonSubsequence(self, text1, text2):
        m=len(text1)
        n=len(text2)
        dp = [[0] * (n + 1) for _ in range(m + 1)]
        for i in range(1, m + 1):
            for j in range(1, n + 1):
                if text1[i - 1] == text2[j - 1]:
                    dp[i][j] = dp[i - 1][j - 1] + 1
                else:
                    dp[i][j] = max(dp[i - 1][j], dp[i][j - 1])
        return dp[m][n]
```

Saved

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2 | Case 3

Input

text1 =