VIRAJ VORA 23107007B SY BTECH CE
DAA LAB 6

NAME: _____ STD. _____ DIV. _____

Page:
Date:

Experiment 1 → Longest common subsequence of grades among students.

* Algorithm

```
function LCS(str1, str2):
    Initialize 2D list 'dp' of size (len(str1)+1, len(str2)+1) to all elements 0
    i → from 1 to len(str1)
        j from 1 to len(str2)
            if str1[i-1] == str2[j-1]:
                dp[i][j] = dp[i-1][j-1] + 1
            else
                dp[i][j] = max(dp[i-1][j], dp[i][j-1])

    Initialize a list 'lcs_seq'
    i = len(str1), j = len(str2)

    while i > 0 and j > 0
        if str1[i-1] == str2[j-1]:
            Append str1[i-1] to lcs_seq
            decrease i and j by 1
        else if
            dp[i-1][j] > dp[i][j-1]
            i--
        else
            j--

    return reverse(lcs_seq)
```

NAME: _____ STD. _____ DIV. _____ Page: Date:

- Time Complexity
  1. Initializing 2d array → O(len(str1) × len(str2))
  2. Filling the 2d array → O(len(str1) × len(str2))
  3. Constructing the LCS → The backtracking loop runs atmost O(min(len(str1), len(str2))) times

∴ Overall Time Complexity is
2 O(len(str1) * len(str2)) + O(min(len(str1), len(str2)))

- Test Cases

1.

Student ID	Grades
S1	CCBBAA - - - CCCC
S2	DDCC - - - - FFFF
S3	BBFF - - - - CDBB
:	:
S20	ABCC - - - - BCAB

Expected Output: LCS is DDD

2.

Student ID	Grades
S1	AAAB - - - - - BBAA
S2	DDAB - - - - BBFF
S3	ABCD - - - - DDAA
:	:
S20	BBAA - - - - CDBC

NAME: STD. DIV. Page: Date:

Expected Output → BBBBC

3] Student ID — Grades
S1 — FFBC . . . FFBB
S2 — ABBC . . . AACC
S3 — DDAB . . . DDCC
⋮
S20 — CDFF . . . LBBBB

Expected Output → CBBB

4] Student ID — Grades
S1 — DDAA . . . CD$@
S2 — BCCFFAAB
S3 — CCAB . . . DAA
⋮
S20 — CCFF . . . FF1A

Expected Output → Error (special char used)

5] Student ID — Grades
S1 — BCAA . . . AB
S2 — AAB . . . DAA
S3 — DDC@1 . . . DAB
⋮
S20 — CDAB . . . BC

Expected Output → Error (Seq length less
40 and special char use

NAME: STD. DIV.

Page:
Date:

6] Student ID — Grades

S1 — CCCC - - - FFIA
S2 — CCDD - - - BDD
S3 — CCC - - - - @ABA
:
S20 — ABAB - - - ABCD

Expected Output -> Error (string length less than 40 and special character used)

Experiment 7 -> Matrix Chain Multiplication optimization

* Algorithm

```
function mcm(n, arr[])
  if n < 2
    return "error: there must least least two matrices"
  if arr.length is not N+1
    return "error: dimension array length must be N+1".
  for each d in arr:
    if d <= 0
      return "error: matrix dimension must be +ve values"
  initialize dp[] of size N x N with all elements set to 0.
  for l from 2 to n-1
    for i from 1 to n-l
      j = i + l - 1
```

NAME: _____ STD. _____ DIV. _____ Page: Date:

3)

```
dp[i][j] = ∞
for k from i to j-1:
    q = dp[i][k] + dp[k+1][j] + arr[i-1] * arr[k] * arr[j]
dp[i][j] = min(dp[i][j], q)
return dp[1][n-1]
```

- Time Complexity
  1. Outer loop (l) → 2 to N-1 so O(n)
  2. Middle loop (i): for each l, i from 1 to n-l, so O(n)
  3. Inner loop (k): O(n) → for each pair i, j, k runs from i to j-1

  Total Time Complexity after combining these nested loops is O(n³)

- Test Case
  1. ([7,5,4,6,7,8], 5)
     Expected Output: 504
  2. ([3,7,5,10,15], 4)
     Expected Output: 255
  3. ([2,4,5,6,8], 4):
     Expected Output: 100
  4. ([4,8,6,7,9], 4):
     Expected Output: 360
  5. ([4,3,6,4,8], 4):
     Expected Output: 156

NAME: _____ STD. _____ DIV. _____

Page:
Date:

6. ([3,7,4,7,5])
Expected output: Error

7. ([2,5,6], 1)
Expected Output: Error

8. ([10,20,30], 2)
Expected Output: Error

9. ([10,-20], 1)
Expected Output: Error

10. ([0,20,10], 2)
Expected output: Error

Conclusion
Here we have studied the SOLID principles of programming and also studied Algorithms to find longest Common Subsequence and Matrix Chain multiplication.