Name Vansh Sharma

Class A6\_B3\_45

TASK 1: Find the similarity between the given X and Y sequence.

X=AGCCCTAAGGGCTACCTAGCTT

Y= GACAGCCTACAAGCGTTAGCTTG

Output: Cost matrix with all costs and direction, final cost of LCS and the LCS.

Length of LCS=16

text1 = "AGCCCTAAGGGCTACCTAGCTT"

text2 = "GACAGCCTACAAGCGTTAGCTTG"

matrix = [["" for x in range(len(text2))] for x in range(len(text1))]

for i in range(len(text1)):

for j in range(len(text2)):

if text1[i]==text2[j]:

if i==0 or j==0:

matrix[i][j] = text1[i]

else:

matrix[i][j] = matrix[i-1][j-1]+text1[i]

else:

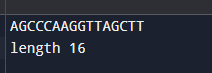
matrix[i][j] = max(matrix[i-1][j], matrix[i][j-1], key=len)

v = matrix[-1][-1]

print(v)

print("length", len(v))

output



TASK-2: Find the longest repeating subsequence (LRS). Consider it as a variation of the

longest common subsequence (LCS) problem.

Let the given string be S. You need to find the LRS within S. To use the LCS framework, you

effectively compare S with itself. So, consider string1 = S and string2 = S.

Example:

AABCBDC

LRS= ABC or ABD

Code:

s1="AABEBCDD"

s2=s1

n = len(s1)

m = len(s2)

matrix = [["" for x in range(len(s2))] for x in range(len(s1))]

for i in range(0, n):

for j in range(0, m):

if s1[i-1] == s2[j-1] and i!=j:

matrix[i][j] = s1[i-1] + matrix[i-1][j-1]

else:

matrix[i][j] = max(matrix[i-1][j], matrix[i][j-1], key=len)

v=matrix[-1][-1]

h=v[::-1]

print(h)

output:



Leetcode:

