## DAA LAB-6

Page No.

1. Longest common subsequence:

1/ Templementing longest common subsequence to find longest subsequence between 2 strings.

11 Input: String 51, 52, and 2 pointers mand 1

11 Output: length of the longest common subsequence

def les (51, 52, m,n):

if m == 0 on n == 0

neturn 0

if memo [m][n]! = -1

neturn memo [m][n]

if s | [m-1] = -s2[n-1]: memo [m][n] = 1 + |cs(s|, s2, m-1, n-1)Aetum memo [m][n]

else:

memo[n][n] = man (lcs(s1, s2, m, n-1))

lcs(s1, s2, m-1, n))

return memo(mJ[n]

Time complenit:

a) Brute Force

- Generate all subsequences

Ton string 51 of lengthm m (m (n) there
are 2<sup>m</sup> possible subsequences. This is because
the characters of 51 can be either included on
encluded in a subsequence leading to 2"combination

- Checking each subsequence

Too each subsequence generated from slave

need to check if it is a subsequence of 52

of length n. This is done in times = 0(n)

:. Time conflerity = O( 2 min(m,n), man (m,n))

= 0(2 xn) y

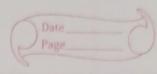
b) subproblems

|     | Page No. Date  |
|-----|--|
|     | for 'n' Strings'.  |
|     | a remarked property of the second second   |
|     | Algorithm Find LCSMultiple (strings)   |
|     | (Computer the LCS comong in Straings (Contract: Array of in straings (Contract: LCS of on straings)  |
|     | Infut: Amay of n strings   |
|     | Coughe: LCS of on strings  |
|     | man - seg = ""   |
|     | for i=0 to n-1:  |
|     | LCS = SCi7   |
|     | fon j = 0 + n - 1  |
|     |  |
|     | LCS= FindLCS(LCS, S[j])  |
|     | ILCS == "  |
|     | broak man 100 un 100 un  |
|     | man-seq = man (man-seq, LCS, vey = length  |
|     | Actum man-seg.   |
|     |  |
|     | Time Complenity:   |
| 1 1 | 1) Thomas are O 1 21 and 100m O had 4  |
|     | 1) There are 2 loops, one from 0 to 1.1, other   |
|     | 2) In the inner most operation, commuting LCS is B(12)   |
|     | 2) In the inner most operation, computing LCS is B(12) in the worst case, assuming an average string   |
|     | Length of L.   |
|     | 3) Hence total time complexity is  |
|     | $T = \frac{5}{5} = $ |
|     | 1-0 j=0  |
|     | 7  |
|     |  |

Page No.

| A      | Test cases (les) · Positive   |
|--------|---|
| 0      | TO GRADES (G) ID CARROLES  SI COPPARCC SI DOFFBCCC  12 DOCCABFF S2 ABBBBCCC  53 FF BBBBCC S3 AAABBCDD |
| test d | Output: DDD Output: BBCCC   |
| 0      | In chade & In Chades  SI AABB AABB SI ABBCCC DV  SI POCC AABB SI AAAB CCDD  SI ABBC BCBC SI AAAA BBBB |
| NA GA  | 520 BBAA. BBCC S20 ABBCCCDD OUTPUT: BBBBC Output: BCCD  |
| (3)    | ID Mades  SI BCFF AABB  |
|        | 52 CCDDABBC<br>53 BBCCDDFF  |
|        | Output: BBCCC   |
|        |   |

Negagine Test Cases Grade I) mada BBCC...CCFF 51 AIB(W...CCDD AAAB ... Be CO ABBC -- c@ Q4 ABBE ... CODD 520 AABB ... CCDD 520 ABCC. AABB than 40 and special charactery Empeded out Put: special chandes Grades Gnades AFBD ... FECK AABC ... CBBD ABBC CCDD ACCA ... ABBK author: Invalid grades Enflected output: string length less than 40 Grades A1 B2... C3 D4 51 A2B8 -- CCDD 520 A9BB ... CCPF Enpeted output: Numbers used in string

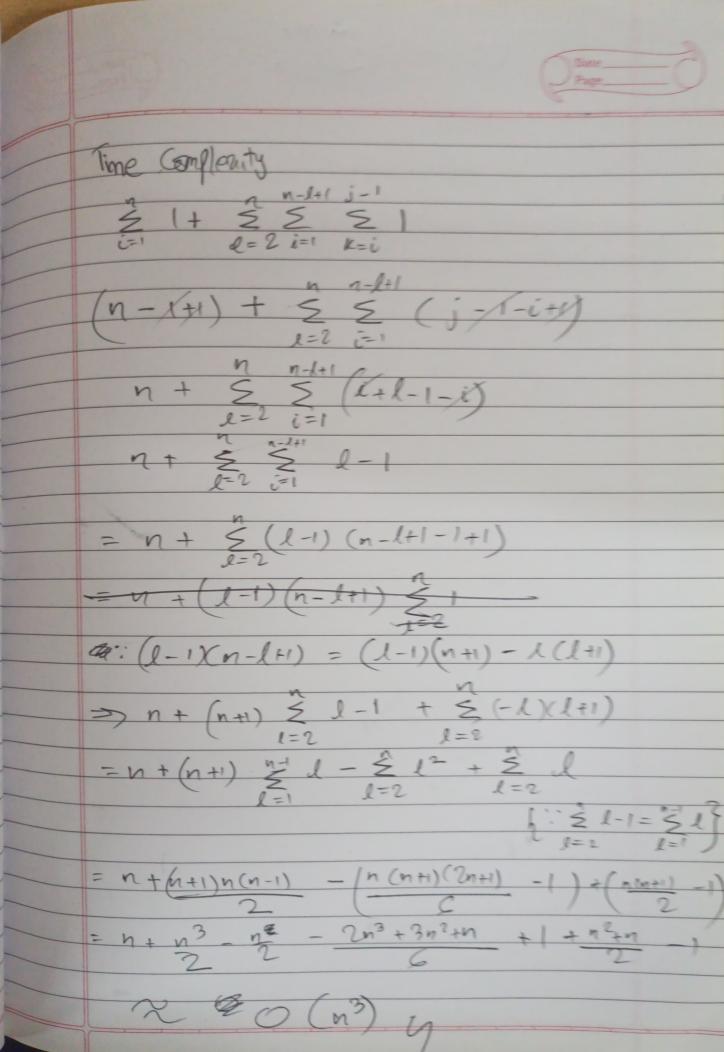


Hatrin Chain Multiplication (P) Two implement Matrin Chain Multiplication to find most effecient way to multiply in' matrices:

Martines: List containing size of matrin 2 2

pointers i & matrin chair.

Moutput: The minimum steps of matrin chair. del mem (p) n = p. length - 1 let m[1...n, 1...n] and s[1,..n-1, 2...n] be new tobles for i= 1ton fon j = 2 to n fon i = 1 ton-l+1 fon i = i+l-1if q < m[i,j] m[i,j] = 9 5[i,j]= K return m and 3



Test coses: mcm [ ([7, 5,4,6,7,8],5) -> ofp: 504 2] ( [3, 7, 5, 10, 15], 4) -> 0/p: 255 3 ([2,4,5,6,8],4)->0/P:100 4] ([4,8,6,7,9],4)->0/p:360 5] ([7,3,6,4,8],4) -> 0/P:156 6] ([3,7,4,7,5]) -> O/P: FMON 7] ([2,5,6],1) -> 0/p: eEmon 8] ( E 10, 20, 30], 2) -> Off: Emon 9] ([10,-20], 41) -> 0/p: EMON 0] ([0,20,10],2) -> 0/p: Enron

Conclusion:

Pence we have studied the SOLID

principles of programming and also

studied Algorithms to find longest Smmon
Subsequence and matria chain multiplication.