	classmate
BY6 P	ate

	14/06/2023
	station of the second
<u>Q</u> I	Longest common subsequence
	. 30 M coc.
	i/b → SI= "abcde"
	S2 = "ace"
b	0/b-1 3 (ace is the longest common
3 '	Subsequence)
1-1-1-13	S = abc 1 may 1 x p m
-	Subsequence of above string can be a, b, c, ab, bc, ac, abc, "" 4 empty string
	ab, bc, ac, abc, ""
	5 empty string
310	FIRST CONTRACTOR AND
$\in x_{1}$	abc
	a regional military and
	Ь
	Standard of the property of the property of the standard of th
	ab
	bc Cd
COLLY .	ac ad ad
	acd
	Company Cul Cons
	Common subsequences are a, c, and ac.
	Here ac is the longest subsequence here.
	Approach
	text1 - abcde
	text2 + ace
	C;
	text1 [i] = = text2[j]. 3 char matches
	rest of both strings, recursion will handle.
	will handle
	Mec(i+1,j+1)
	Scarnieu with ca

Scarined with Cam

But what if character does not match. text1 - abcde tesce - vine Simply take max (Hec (i+1,j), Hec(i,j+1)); exclude 1st string char String char The above approach is 2-pointer approach. Code int solve Rec (string a, string b, int i, int j) { 1/ Base case (String completely traversed) if (i = = a · length()) return Oi // Base case if (j == b length ()) return Oi // Match if (a[i] = = b[j]) return 1 + solve Rec (a, b, i+1, j+1); // Mismatch else { return 0 + max (Solve Rec (a, b, i, j+1), solve Rec (a,b, i+1,j)) 11 Top-down approach (Gave TLE if 4 wasn't used) int solve Top Down (String & a, string & b, inti, int j, vector <vector <int>) &db) {

ocanneu wini can)

_	
	//Base case
	if (i = = a length ())
	return 0;
	if (j b·length ())
	return Oi
	// Step 3: Check ans exists in dp or not
11));	if (ab(i][j] 1 = -1)
	return de[i][j]
1	// Match case
W	int ans = 0;
	if (a [i] = = b[j])
	ans = 1+ solve Top Down (a, b, i+1, j+1, dp);
	//Mismatch case
	else {
31.	ans=0+max (solve Top Down (a,b,i,j+1,dp),
	solve Top Down (a,b,i+1,j,dp));
	1 3
	//Step 2: Store ons in dp away
	db[i][j] = ansi
	return de [i][j]i Amalda
	3
Note -	dp array created in main () as : vector < vector < int >> dp (a length (), vector <int> (b · length (), -1));</int>
	vector (vector (int)) db (0 length () vector
	<int> (b. length (), -1));</int>
	// Bottom up approach
	int solve Tab (string a String b) of
	int solve Tab (string a, String b) { // Step 1: Create dp away
	vector (vector (int)) db (a. lenation)
Ĭ i 2. :-	
ا ز	1/ Step 2 . Observe have care of 1.
	// Already Landled in initialization step
	o step
	Scarnieu with Cal

ocanneu with odm

```
// Step 3: Reverse flow of top-down
   for (int i = a length () -1; i>=0; i--){
        for (int j = b length () -1 jj > = 0 j - -) {
            // match case
            Int ans = Oj
            if (a[i] = = b[j])
                ans = 1 + db [i+1][j+1];
          //Mismatch case
          else 1
             ans = 0 + max (db[i][j+1], db[i+1](j]);
        dp[i][j] = ansi
    retwin db [0][0];
* Space optimization possible or not?
  Yes, space optimization is possible
1) Create 2,1 D arrays
   Vector (int) curi (b length () +1,0);
    Vector Kint) next (b.length ()+1,0);
  Replace of [i+D--with next[--] and
  dp [i] -- with cur [--]
  Shifting - next = curr as we are going upwards.
  Longest palindromic subsequence
   text1 = 1/b String
  text2 = reverse of 1/p string
  Apply longest common Subsequence on text1
   and text 2 Strings.
```

Scarined with Carif

(Very famous question)

Q3	Edit distance
	We will be given 2 words in 1/p and
	we need to abbly minimum operations
	such as insert, delete or replace character
	to make the second word.
	Conversion
	WHATHAMAN W2
	Li di
$Ex \rightarrow$	110730
CIFFILM	Thi Jico Ato som + c = mp
	Here 1st characters are matching. Hence simply recursive call for i+1 and j+1.
	Simply recursive call for i+1 and j+1.
	Insertion case
	mhorse (Lastan dans)
	inserted
,	Hence recursive call for a and j+1 as now
	Ist character of both strings are matching.
	Deletion case
William -	Korsel (Japan) a) Haran Charle modern
and subspects of the su	Jacobe - Delan Hiller Caron Del
	Hence recursive call for i+1 and j.
wards	propries and the mount of the
	Replace case
	mhorse man man sour sime who is a
	mse
	Now character are matching and hence
3	recursive call for it and j+1.
14	of an arrangement there was a facility
2 1.	and the same of th

Scarined with Cam

	Code de la little de la laconstant de laconstant de la laconstant de laconstant de la laconstant de laconstant de laconstant de la laconstant de laconstant de la laconstant de laconstan
	i if (a file begins begins for an
(4t)	int solve Rec (string & a , string & b , int i , int j) {
	// Base case
	$if (i = a \cdot length ())$
196 - S. 196	retwin b. length ()-jj
1/22	If $(i = b \cdot length())$
112	return a length ()-ii
Mah.	1/match case
	wint and = 0 jums of some and
	$if (0 \Gamma i) = = b(1)$
	ans = solve Rec (a, b, L+1, j 11)
	//Mismatch - Operations to be performed
	int insert = $1 + SolveRec(a,b,i,j+1);$
	A = A + A + A + A + A + A + A + A + A +
	$1 + cnive Rec (950) \times (77)$
	and (incest a min (aelected) replaces)
	3 We need to apply the
	retwin ans i operations
	3
	// Top down approach (china & a. String & b. int i) int i) i
	lint solve Top Down (Siring ag)
	1/Base case
	$i(C) = -Q \cdot length(Q)$
	return bilength () j
	$if (j = b \cdot length ())$
	return a length of x
	if (j = = b. length () - i ; Yeturn a length () - i ; // Step 3: Check if answer already exists in dp
	10/11/11/11/11/11
	retwin de [i][j]

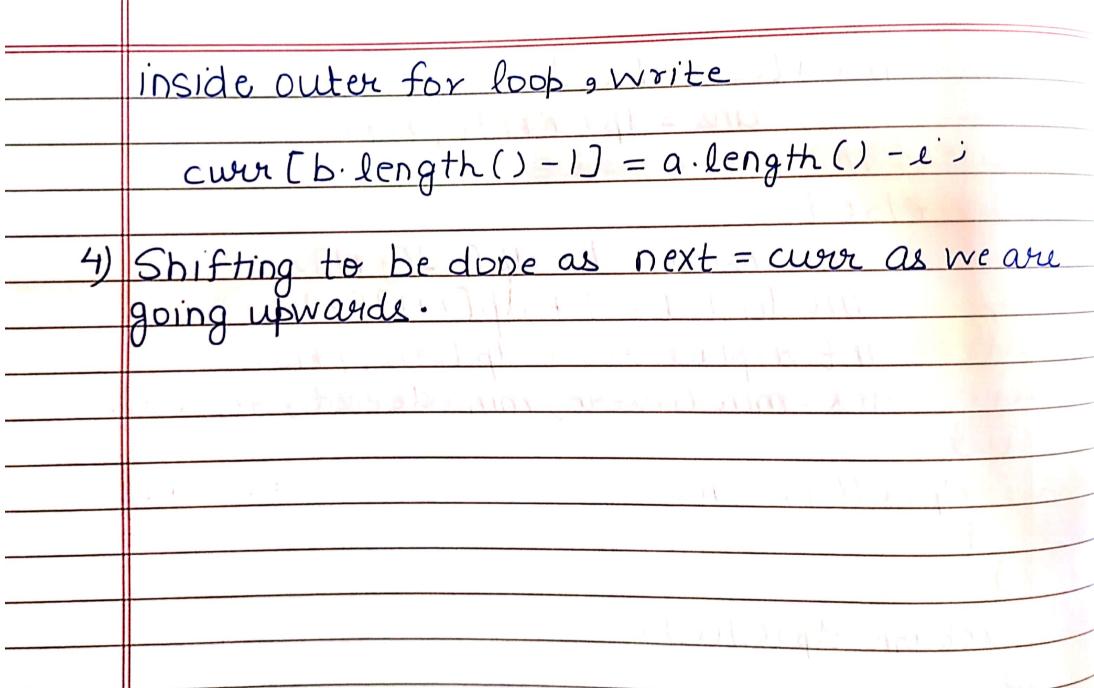
1	Page(')
	int ans = 0
	if (a(i) = b(j))
1 171	ans = solve Top Down (a, b, i+1, j+1, dp);
	3
	11 mismatch
	else {
	int insert = 1+ solve Top Down (a, b, i, j+1, db);
	int deleted = 1 + Solve Top Down (a, b, i+1, j, db);
	int replace = 1 + solve Top Down (a, b, i+1, j+1, dp);
	ans=min (insert, min (deleted, replace));
	//Step 2: Store and in dp averay
_	aplij = ans;
	retwin de [i][j]
. ,	
-	// Bottom up approach
111	int solve Tab (string & a. string & b) { //Steb 1: (rept. db a. string & b) {
	11Step 1. (reate de avoiay
	//Step 1: (reate dp avriay vector (vector (int >) dp (a length () + 1, vector (int > (b length () + 1,
	vector Kint > (b. length () + 1, 0)); //Step 2: Observe base (C) (a length () + 1, 0));
	for lint i - a in it has care of tob down
li toi	$db \Gamma O \cdot land () = b \cdot length () \cdot 1 + +) {$
	//Step 2: Observe base case of top down for (int j = 0 : j < = b · length () : j + +) { dp [a · length ()][j] = b · length () - j : 3
	for (int i = 0; [<=a.length(); i++) { dp[i][b.length()] = a.length()-i;
	dp[i](b·length())i++){
	3 (length ()-1)
	1 // 57/21 / 0 1)
	for (int i = a length () - 1;
	for (int i = a · length () - 1 ; i > = 0; i) { for (int j = b · length () - 1; j > = 0; j) { // Match case int ans = 0;
	Match case
	int ans = 0;
	Scarined With Car

Scarifica with cam

Scarineu with Caril

	$if(\alpha[i] = b[j])$
	ans = $d\beta (i+1)[j+1]j$
	// Mismatch case
	else {
	int insert = 1+dp(12)(j+1)
	int deleted = 1 + dp(i+1)[j];
15	int replace = 1 + dp [i+1](j+1);
	ans = min (insert, min (deleted, replace));
	3
· seq	dp[i][j] = ansi
	3
	3
	return dp [0](0];
	3
	· /
*	Space optimization is possible or not?
	Yes space optimization is possible
	(reate 2 ID arrays Vector Kint) Curv (b length () + 1,0); Vector Kint) next (b length () + 1,0);
	Vector Rint Coul (b length ()+100);
	Vector Since Co
21	Replace dp[i+1] with next [] and dp[i] with curr [].
3)	ALCIA with cour []
3)	Base case modific tions
 	for (int j = 0) j <= b length (), j+T)
	next [J] = b.length() -J)
4	For (int j = 0) j <= b length() ; j++) next [j] = b length() - j; r Can do mistake here and have cove is tricky here as
<u>*</u>	Now and base case is tricky here as we don't have all the rows. Hence in Step-3
	We don't have all the ours





Scarineu Willi Cail