

DP-5

Length of Longest Common SS problem

$\frac{n}{2} \rightarrow x = " \underbrace{a \ b \ c \ d}_{\text{common}} \ x \ y \ z "$

$\frac{n}{2} \rightarrow y = " \ x \ y \ z \ \underbrace{a \ b \ c \ d}_{\text{common}} "$

$\rightarrow 4$

$\rightarrow abcd$

Max  
 $\Rightarrow$  length of longest common sub-sequence.

length  $\rightarrow 'n'$

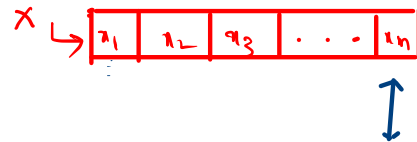
$\downarrow$

$2^n$

\* length

① optimal ss:

length:-11



$$X = (x_1, x_2, \dots, x_m)$$

$$|X| = m$$



$$Y = (y_1, y_2, \dots, y_n)$$

$$|Y| = n$$

$x_m$  v/s  $y_n$

equal

not equal

$$LCS(x_i, y_j) = \begin{cases} 0 & ; i=0 \text{ or } j=0 \\ 1 + LCS(x_{i-1}, y_{j-1}) & ; x_i = y_j \\ \max(LCS(x_i, y_{j-1}), LCS(x_{i-1}, y_j)) & ; x_i \neq y_j \end{cases}$$

$1 + (m-1, n-1)$

$\max \{ (m, n-1), (m-1, n) \}$

longest



## ① Recursive code-

```
//m:: length of array x
//n:: length of array y
function LCS(x[],y[],m,n)
{
    if(m==0 || n==0)
        return 0;
    if(x[m-1]==y[n-1])
        return 1+LCS(x,y,m-1,n-1)
    else
        return max( LCS(x,y,m,n-1), LCS(x,y,m-1,n) )
}
```

## ② TD - DP:

↳ 1. Recursive call.

↳ 2. Memoization → create a table.

```
//TD-DP Code for LCS problem
//m:: length of array x
//n:: length of array y
dp[m+1][n+1]={-1} // set all values with -1
function LCS(x[],y[],m,n)
{
    if(m==0 || n==0)
        return 0;
    if(dp[m][n]!=-1)
        return dp[m][n]
    if(x[m-1]==y[n-1])
        return dp[m][n]=1+LCS(x,y,m-1,n-1)
    else
        return dp[m][n]= max( LCS(x,y,m,n-1), LCS(x,y,m-1,n) )
}
```

② Bottom-up - DP:

↳ ① loops

↳ ② memoization

$x[] \quad \left. \vphantom{x[]} \right\} \rightarrow |x| = m$   
 $y[] \quad \left. \vphantom{y[]} \right\} \rightarrow |y| = n$

$dp[m+1][n+1]$

x: <sup>0</sup>a <sup>1</sup>c <sup>2</sup>b <sup>3</sup>c <sup>4</sup>f → 5      m = 5

y: <sup>0</sup>a <sup>1</sup>b <sup>2</sup>c <sup>3</sup>d <sup>4</sup>a <sup>5</sup>f → 6      n = 6

$(m+1) \times (n+1)$

6 x 7

→ y (n)

dp →

	0	<sup>1</sup> a	<sup>2</sup> b	<sup>3</sup> c	<sup>4</sup> d	<sup>5</sup> a	<sup>6</sup> f
<sup>0</sup>	0	0	0	0	0	0	0
<sup>1</sup> a	0	1	1	1	1	1	1
<sup>2</sup> c	0	1	1	2	2	2	2
<sup>3</sup> b	0	1	2	2	2	2	2
<sup>4</sup> c	0	1	2	3	3	3	3
<sup>5</sup> f	0	1	2	3	3	3	(4)

(m) x

6 x 7

dp →

		1 a	2 b	3 c	4 d	5 a	6 f
0	0	0	0	0	0	0	0
a	0	1	1	1	1	1	1
c	0	1	1	2	2	2	2
b	0	1	<u>2</u>	2	2	2	2
c	0	1	2	3	3	3	3
f	0	1	2	3	3	3	(4)

6x7

X = "a c b c f"  
 Y = "a b c d a f" } ⇒ 4 ✓

a b c f ✓

match

no match

res = "f c b a"

return res.reverse()

a b c f



```
i=m,j=n
temp=""
while(i>0 && j>0)
{
    if(x[i]==y[j])
    {
        temp=temp+x[i]
        i--, j--
    }
    else
    {
        if(dp[i-1][j]>dp[i][j-1])
            i--
        else
            j--
    }
}
print(temp.reverse())
```

//BUP dp code for LCS

function LCS(x[],y[],m,n)

{

dp[m+1][n+1];

for(i=0;i<=n;i++)

dp[0][i]=0

for(i=0;i<=m;i++)

dp[i][0]=0

for(i=1;i<=m;i++)

{

for(j=1;j<=n;j++)

{

if(x[i-1]==y[j-1])

dp[i][j]=1+dp[i-1][j-1]

else

dp[i][j]=max(dp[i][j-1],dp[i-1][j])

}

}

return dp[m][n]

}

m → i

n → j

