

Experiment - 7

Obj :- To find the longest common subsequence of two strings and analyse the time complexity of the algorithm.

Theory :- The longest common subsequence of two strings is the longest subsequence i.e. common to both the strings. A subsequence is a sequence that appears in the same relative date order, & is not necessarily contiguous.

For "AGAGTACB" and "GXTXAYB", the LCS is GTAB.

Algorithm:-

$\text{LCS}(\text{char} * x, \text{char} * y)$

$m \leftarrow x.\text{length}$

$n \leftarrow y.\text{length}$

$L[m+1][n+1] \leftarrow 0$

for  $i$  in  $0$  to  $m$

    for  $j$  in  $0$  to  $n$

        if ( $i == 0$  or  $j == 0$ )

$L[i][j] \leftarrow 0$

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

$L[i][j] \leftarrow L[i-1][j-1] + 1$

        else  $L[i][j] = \max(L[i-1][j], L[i][j-1])$

    end if

Analysis:-

While calculating the LCS, we traverse the  $2 \times$   $m \times n$  array once. Hence, the complexity is

$$\Theta(mn)$$

for  $i$  from 0 to  $m$

No. of Ene.

$$(m+1)$$

for  $j$  from 0 to  $n$

$$m(n+1)$$

$$f(n)$$

$$(m \times n)$$

Hence the complexity is  $\Theta(m \times n)$ .

end for  
end for

index  $\leftarrow L[m][n]$   
char LCS [index + 1]  
LCS [index]  $\leftarrow ''$

i  $\in m$

j  $\in n$

while (i > 0 and j > 0)

if ( $X[i-1] = Y[j-1]$ )

LCS [index - 1]  $\leftarrow X[i-1]$

i--, j--, index --;

else if (L[i-1][j] > L[i][j-1])

i--;

else

j--;

end if

end while

return LCS

end

Result:- The LCS was successfully found, & the complexity of the algorithm was analyzed.