

Experiment - 7

Page No.	
Date	

Aim :- To find the longest common subsequence of two strings and analyze 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 order, & is not necessarily contiguous.

Eg "A G A T A B" and "A X T X A Y B", the LCS is A T A B.

Algorithm :-

LCS (Char * X, Char * Y)

$m \leftarrow X \cdot \text{length}$

$n \leftarrow Y \cdot \text{length}$

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

for i in 0 to m

for j in 0 to n

if $(i == 0 \text{ or } j == 0)$

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

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

Elements: Array, List,

List to Set

Analysis:-

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

$O(mn)$

for i from 0 to m

for j from 0 to n

$f(n)$

No. of Elements

$(m+1)$

$m(n+1)$

$(m \times n)$

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

end for
end for

Page No.	
Date	

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

$i \leftarrow m$

$j \leftarrow n$

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 analysed.