Analysis of Algorithms

[5CS4-05/5IT4-05]

Unit 2: Dynamic Programming

Longest Common Subsequence

Ramakant Soni

Assistant Professor, Computer Science Department,

B K Birla Institute of Engineering & Technology, Pilani

Rajasthan

Longest Common Subsequence

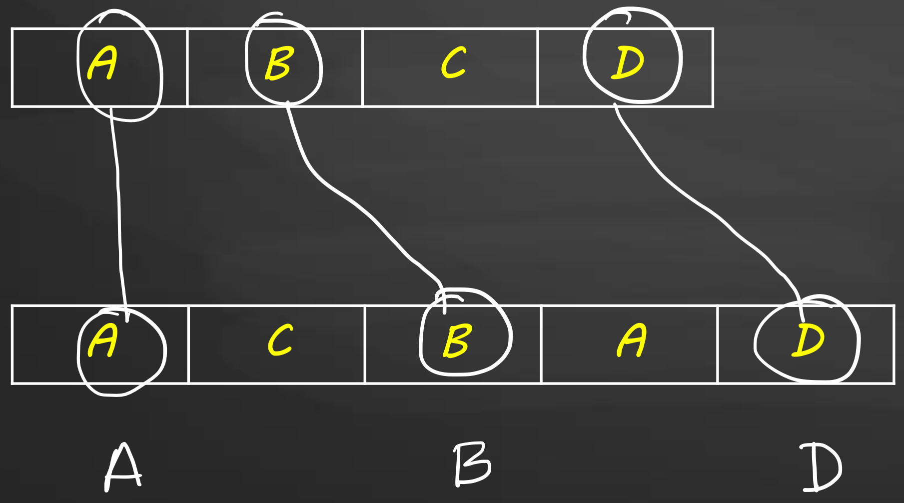
The longest common subsequence (LCS) is defined as the longest subsequence that is common to all the given sequences, provided that the elements of the subsequence are not required to occupy consecutive positions within the original sequences.

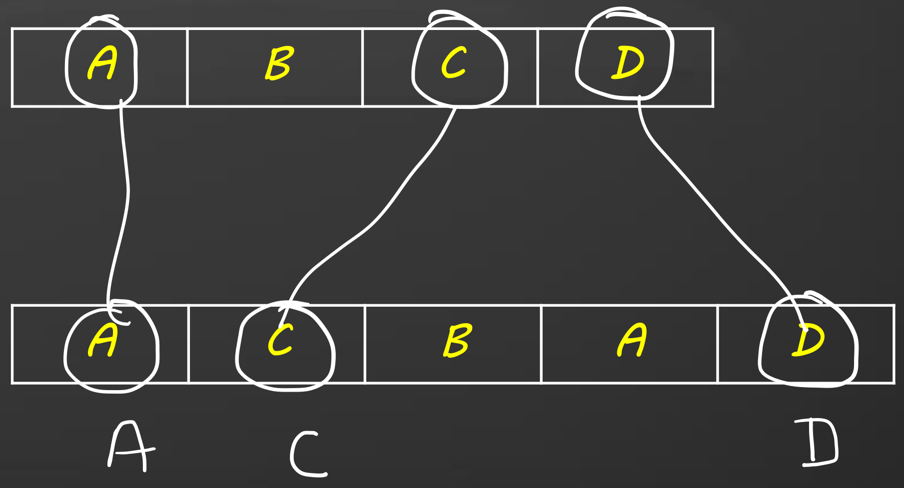
Longest Common Subsequence

For example, consider the sequences (ABCD) and (ACBAD).

They have 5 length-2 common subsequences: (AB), (AC), (AD), (BD) and (CD)
2 length-3 common subsequences: (ABD) and (ACD)
and no longer common subsequences.

So (ABD) and (ACD) are their longest common subsequences.





1 Algorithm to compute LCS length

```
LCS-LENGTH(X, Y)
m = X.length
n = Y.length
let b[1..m, 1..n] and c[0..m, 0..n] be new tables
for i = 1 to m
    c[i, 0] = 0
for j = 0 to n
    c[0, j] = 0
for i = 1 to m
    for j = 1 to n
        if x_i == y_j
            c[i, j] = c[i − 1, j − 1] + 1
            b[i, j] = "↖"
        elseif c[i − 1, j] ≥ c[i, j − 1]
            c[i, j] = c[i − 1, j]
            b[i, j] = "↑"
        else c[i, j] = c[i, j − 1]
            b[i, j] = "←"
return c and b
```

Ramakant Soni, Assistant Professor, BKBIET Pilani

2 Algorithm to print LCS

```
PRINT-LCS(b, X, i, j)
if i == 0 or j == 0
    return
if b[i, j] == "↖"
    PRINT-LCS(b, X, i − 1, j − 1)
    print x_i
elseif b[i, j] == "↑"
    PRINT-LCS(b, X, i − 1, j)
else PRINT-LCS(b, X, i, j − 1)
```

Define c[i, j] to be the length of the LCS of x[1··i] and y[1··j]

Eg :-

	0	1 A	2 B	3 C	4 D	5 B
0						
1 A						
2 C						
3 B						

for $j = 1$ **to** n

  if $x_i == y_j$

    $c[i, j] = c[i-1, j-1] + 1$

    $b[i, j] = \text{“}\nwarrow\text{”}$

  elseif $c[i-1, j] \geq c[i, j-1]$

    $c[i, j] = c[i-1, j]$

    $b[i, j] = \text{“}\uparrow\text{”}$

  else $c[i, j] = c[i, j-1]$

    $b[i, j] = \text{“}\leftarrow\text{”}$

Define $c[\,i, j\,]$ to be the length of the LCS of $x[\,1..i\,]$ and $y[\,1..j\,]$

	0	1 A	2 B	3 C	4 D	5 B
0	0	0	0	0	0	0
1 A	0	↖ 1	← 1	← 1	← 1	← 1
2 C	0	↑ 1	↑ 1	↖ 2	← 2	← 2
3 B	0	↑ 1	↖ 2	↑ 2	↑ 2	↖ 3

for $j = 1$ **to** n
  if $x_i == y_j$
    $c[i, j] = c[i-1, j-1] + 1$
    $b[i, j] = $ "↖"
  elseif $c[i-1, j] \geq c[i, j-1]$
    $c[i, j] = c[i-1, j]$
    $b[i, j] = $ "↑"
  else $c[i, j] = c[i, j-1]$
    $b[i, j] = $ "←"

Given the two sequence of characters, X= < A B C B D A B >, Y= < B D C A B A >
Find out longest common subsequence

Eg 2 :-

		0	1	2	3	4	5	6	7
		∅	A	B	C	B	D	A	B
0	∅								
1	B								
2	D								
3	C								
4	A								
5	B								
6	A								

LCS=

Ramakant Soni, Assistant Professor, BKBIET Pilani

Given the two sequence of characters, X= < A B C B D A B >, Y= < B D C A B A >
Find out longest common subsequence

		0	1	2	3	4	5	6	7
		Ø	A	B	C	B	D	A	B
0	Ø	0	0	0	0	0	0	0	0
1	B	0	↑ 0	↖ 1	← 1	↖ 1	← 1	← 1	↖ 1
2	D	0	↑ 0	↑ 1	↑ 1	↑ 1	↖ 2	← 2	← 2
3	C	0	↑ 0	↑ 1	↖ 2	← 2	↑ 2	↑ 2	↑ 2
4	A	0	↖ 1	↑ 1	↑ 2	↑ 2	↑ 2	↖ 3	← 3
5	B	0	↑ 1	↖ 2	↑ 2	↖ 3	← 3	↑ 3	↖ 4
6	A	0	↖ 1	↑ 2	↑ 2	↑ 3	↑ 3	↖ 4	↑ 4

LCS Length = 4

Ramakant Soni, Assistant Professor, BKBIET Pilani

Given the two sequence of characters, X= < A B C B D A B >, Y= < B D C A B A >
Find out longest common subsequence

		0	1	2	3	4	5	6	7
		∅	A	B	C	B	D	A	B
0	∅	0	0	0	0	0	0	0	0
1	B	0	↑ 0	↖ 1	← 1	↖ 1	← 1	← 1	↖ 1
2	D	0	↑ 0	↑ 1	↑ 1	↑ 1	↖ 2	← 2	← 2
3	C	0	↑ 0	↑ 1	↖ 2	← 2	↑ 2	↑ 2	↑ 2
4	A	0	↖ 1	↑ 1	↑ 2	↑ 2	↑ 2	↖ 3	← 3
5	B	0	↑ 1	↖ 2	↑ 2	↖ 3	← 3	↑ 3	↖ 4
6	A	0	↖ 1	↑ 2	↑ 2	↑ 3	↑ 3	↖ 4	↑ 4

LCS= B D A B

Ramakant Soni, Assistant Professor, BKBIET Pilani

Queries ?