Date: 26/04/21

Black Board

Design and Analysis of Algorithms

Topics:

- Dynamic Programming V
    - THE LONGEST COMMON SUBSEQUENCE PROBLEM

The LCS Problem

- **Input:** Two strings X and Y or lengths m and n respectively.
- **Output:** The Longest Common Subsequence between X and Y

2 Useful Properties of LCSs

① $X = x_1 x_2 \cdots x_m$

$Y = y_1 y_2 \cdots y_n$

abcde

e
de
cde
⋮

$X = X'S$
$Y = Y'S$
$\Rightarrow$ lcs of X and Y is simply the lcs of X' & Y' with S concatenated at the end.

② Focus on single letter suffixes

$|S| = 1$

Case (i) $X = x_1 x_2 \cdots x_m \rightarrow$ match

$x_m = y_n$ $Y = y_1 y_2 \cdots y_n$ $s = x_m = y_n$

$\Rightarrow$ lcs of $X[1 \ldots m]$ & $Y[1 \ldots n]$ is simply the lcs of $X[1 \ldots m-1]$ & $Y[1 \ldots n-1]$ concatenated with s.

Case (ii) $x_m \neq y_n$

$\Rightarrow$ what the lcs is, x_m & y_n cannot be matching in that lcs.

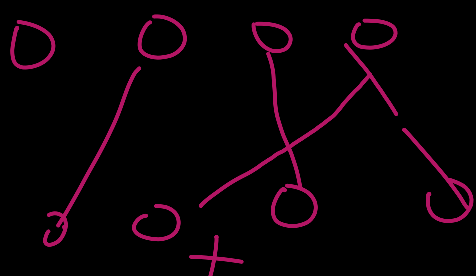
In other words, x_m & y_n together do not play a part in the lcs.

$$X = x_1 \; x_2 \; \cdots \; x_m$$
$$Y = y_1 \; y_2 \; \cdots \; y_n$$

$$x_1 \; x_2 \; \cdots \; x_m$$
$$y_1 \; y_2 \; \cdots \; y_n$$

We pick the best of the two solutions.

$$x_1 \; x_2 \; \cdots \; x_m$$
$$y_1 \; y_2 \; \cdots \; y_n$$



Deriving a recurrence

$lcs[i,j]$:= length of the longest common subsequence between $x_1 x_2 \cdots x_i$ & $y_1 y_2 \cdots y_j$

$$lcs[i,j] = \begin{cases} lcs[i-1,j-1]+1 & x_i = y_j \\ \underbrace{\max\left\{\begin{matrix} lcs[i-1,j] \\ lcs[i,j-1] \end{matrix}\right\}}_{\alpha} & x_i \neq y_j \end{cases}$$

$x_1, x_2 \cdots x_{i-1}$ x_i

$y_1, y_2 \cdots y_{j-1}$ y_j

$x_1, x_2 \cdots x_{i-1}$ x_i

$y_1, y_2 \cdots y_j$

$x_1, x_2 \cdots$ x_i

$y_1, y_2 \cdots y_{j-1}$ y_j

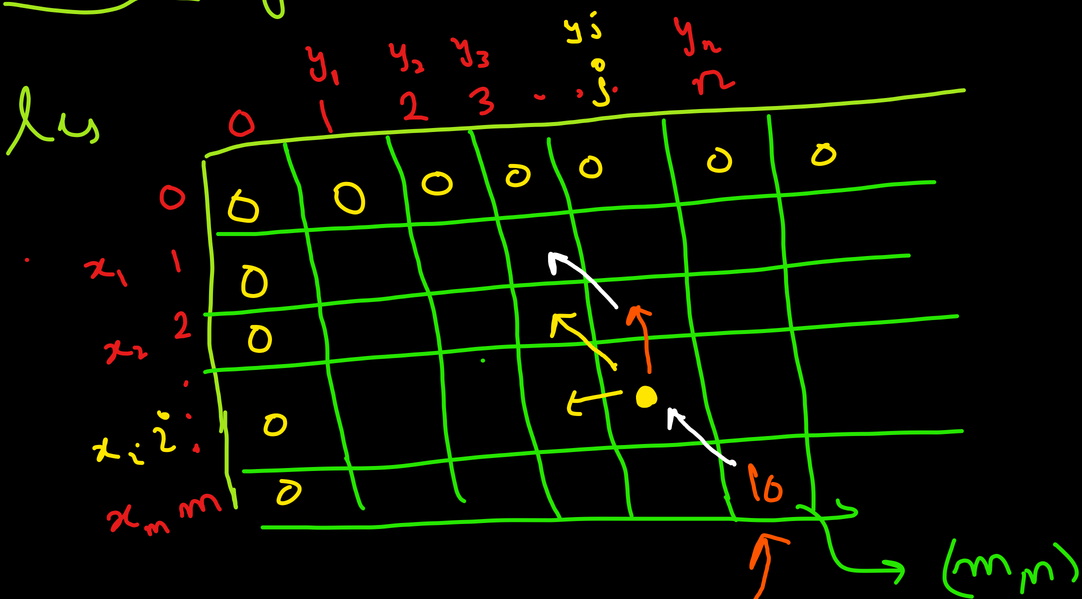
$$lcs[0,j] = lcs[i,0] = 0$$

Define an array P

$$P[i,j] = \begin{cases} (i-1, j-1) & x_i = y_j \\ (i-1, j) \text{ if } \alpha = lcs[i-1,j] & x_i \neq y_j \\ (i, j-1) \text{ if } \alpha = lcs(i,j-1) & x_i \neq y_j \end{cases}$$

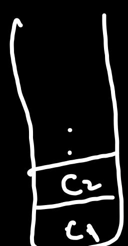
$$\alpha = \max\left\{\begin{matrix} lcs(i-1,j], \\ lcs[i,j-1] \end{matrix}\right.$$

Auxilliary Storage



$P[i,j] = (i-1, j)$

$P[m,n]$



→ pop the lcs from the stack

x= SPRINGS

y=PINES

lcs

P

		A	B	C	D	A
	0	0	0	0	0	0
B	0	0	1	1	1	1
E	0	0	1	1	1	1
A	0	1	1	1	1	2
D	0	1	1	1	2	2
C	0	1	1	2	2	2
A	0	1	1	2	2	3

→ m,n

BDA

B
D
A

3

$$\boxed{T(m,n) = O(mn)} + \max(m,n)$$