

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

$$X = x_1 x_2 \dots x_m$$
$$Y = y_1 y_2 \dots y_n$$

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

abcde
←
e
de
cde
:

② Focus on single letter suffixes
 $|S|=1$

Case (i)

$x_m = y_n$

$$\left. \begin{array}{l} X = x_1 x_2 \dots x_m \\ Y = y_1 y_2 \dots y_n \end{array} \right\} \begin{array}{l} \rightarrow \text{match} \\ s = x_m = y_n \end{array}$$

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

Case (ii)

$x_m \neq y_n$

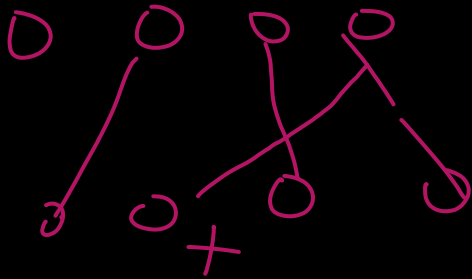
\Rightarrow what the lcs is, x_m & y_n cannot
be matching in that lcs.

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

$$\begin{array}{l}
 X = \begin{bmatrix} x_1 & x_2 & \dots & x_m \end{bmatrix} \\
 Y = \begin{bmatrix} y_1 & y_2 & \dots & y_n \end{bmatrix}
 \end{array}$$

(Note: In the original image, the last elements x_m and y_n are circled in red, and a vertical line separates the two matrices. A checkmark is placed below the first matrix.)

We pick the best of the two solutions.



Deriving a recurrence

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

$$lcs[i, j] = \begin{cases} lcs[i-1, j-1] + 1 & x_i = y_j \\ \max \begin{cases} lcs[i-1, j] \\ lcs[i, j-1] \end{cases} & x_i \neq y_j \end{cases}$$

$x_1, x_2, \dots, x_{i-1}, x_i$
 $y_1, y_2, \dots, y_{j-1}, y_j$

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

$x_1, x_2, \dots, x_{i-1}, x_i$
 y_1, y_2, \dots, y_j

Define an array P

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

x_1, x_2, \dots, x_i
 y_1, y_2, \dots, y_j

$$\alpha = \max \begin{cases} lcs[i-1, j] \\ lcs[i, j-1] \end{cases}$$

Auxiliary Storage

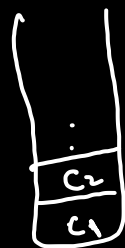
lcs

| | | 0 | 1 | 2 | 3 | ... | j | ... | n |
|-------|---|---|---|---|---|-----|---|-----|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| x_1 | 1 | 0 | | | | | | | |
| x_2 | 2 | 0 | | | | | | | |
| ... | | | | | | | | | |
| x_i | i | 0 | | | | | | | |
| ... | | | | | | | | | |
| x_m | m | 0 | | | | | | | |

Diagram illustrating the auxiliary storage table for the Longest Common Subsequence (LCS) problem. The table is a grid with rows indexed by $x_1, x_2, \dots, x_i, \dots, x_m$ and columns indexed by $0, 1, 2, 3, \dots, j, \dots, n$. The value 0 is written in the first row and first column. A yellow dot is placed at the intersection of row x_i and column j . Arrows point from this dot to the cells $(i-1, j)$, $(i, j-1)$, and $(i-1, j-1)$. A green arrow points from the cell (m, n) to the label (m, n) .

$$\underline{\underline{P[i, j] = (i-1, j)}}$$

$$P[m, n]$$

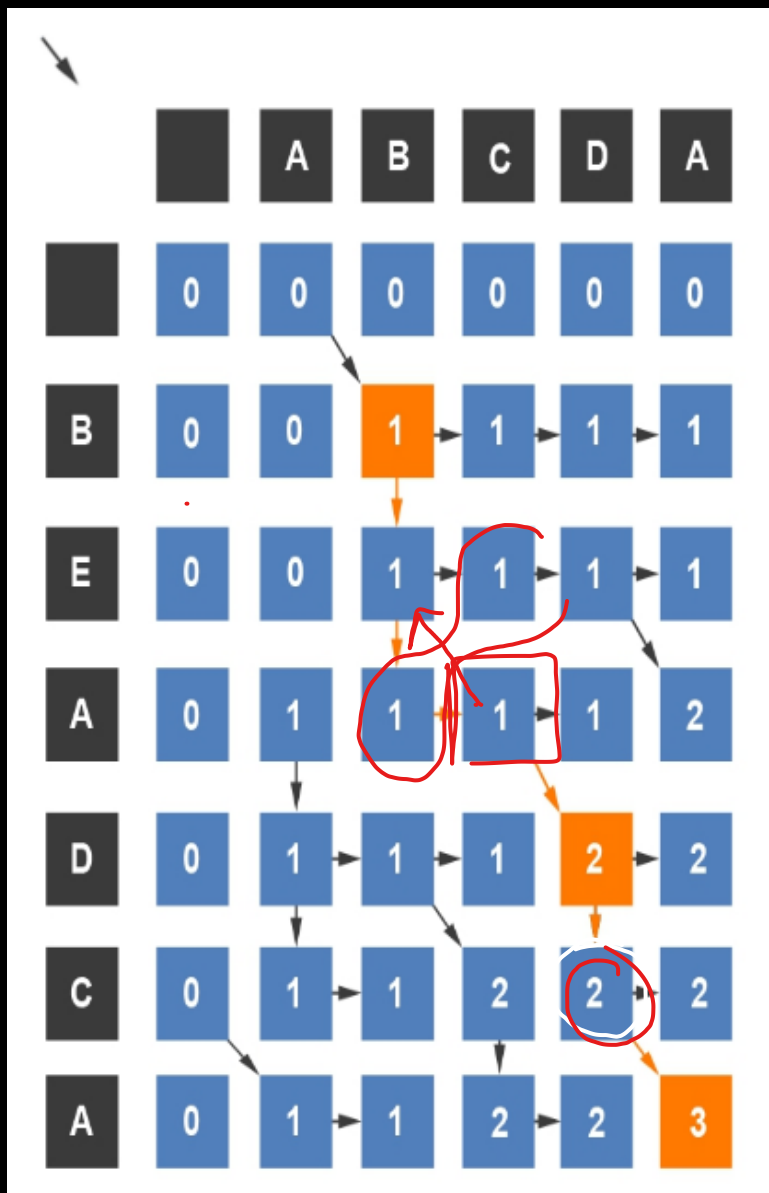


→ pop the lcs from the stack

$x = \text{SPRINGS}$
 $y = \text{PINES}$

Ques

P



BDA

| |
|---|
| B |
| D |
| A |

→ 0, 1, 1

3

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

