**Longest Common Subsequence (LCS):**

**Input:** Strings \( X = x_1, \ldots, x_n \) \& \( Y = y_1, \ldots, y_n \)

**Goal:** Length of longest string that is a subsequence in both \( X \& Y \)

**Example:**

\[
X = \underline{Bc}D\underline{BCDA} \\
Y = \underline{ABEC}B\underline{AA}
\]

**Solution:** 4 corresponding to \( BCBA \)

**Application:** UNIX `diff`

**Step 1: Define subproblem.**

**Attempt 1:** Consider prefixes.

For \( 0 \leq i \leq n \),

\[
L(i) = \text{length of LCS in } X_1, \ldots, x_i \text{ with } Y_1, \ldots, y_i
\]
Can we express $L(i)$ in terms of $L(1), \ldots, L(i-1)$?

Earlier example: BCDBCDAB

For $L(7)$: $X_7 = Y_7$ so might as well match

& take best of $L(6)$

So $L(7) = 1 + L(6)$

But what if $x_i \neq y_i$? E.g. $L(6)$.

Might use $x_i$ or $y_i$ but not both

then left with diff. length strings.

Attempt 2: For $0 \leq i, j \leq n$,

let $L(i, j) = \text{length of LCS of } X_i, \ldots, X_i$

with $Y_j, \ldots, Y_j$
Base cases: \( L(i, 0) = 0 \) & \( L(0, j) = 0 \)

For recurrence, two cases: \( x_i = y_j \) or \( x_i \neq y_j \)

Suppose \( x_i = y_j \):

LCS ends with \( x_i = y_j \) then:

\[
L(i, j) = 1 + L(i-1, j-1)
\]

otherwise can add \( x_i = y_j \) to make it longer. If \( x_i \) matched with \( y_l \) for \( l < j \), then can match to \( y_j \) for same length.

Hence, \( L(i, j) = 1 + L(i-1, j-1) \).

Suppose \( x_i \neq y_j \):

Either \( x_i \) not matched &/or \( y_j \) not matched (can't both be matched)

So, \( L(i, j) = \max \{ L(i-1, j), L(i, j-1) \} \)
Summarizing:

\[
L(i,j) = \begin{cases} 
0 & \text{if } i=0 \text{ or } j=0 \\
1 + L(i-1,j-1) & \text{if } X_i = Y_j \\
\max \{ L(i-1,j), L(i,j-1) \} & \text{if } X_i \neq Y_j 
\end{cases}
\]

\[
L = \begin{array}{c}
\text{fill table } L \text{ row-by-row} \\
\text{to get } L(i,j): \text{ use } L(i-1,j), L(i-1,j-1), L(i,j-1)
\end{array}
\]

LCS(X, Y): X = x_1, \ldots, x_n \text{ & } Y = y_1, \ldots, y_m

For \( i = 0 \rightarrow n \), \( L(i,0) = 0 \)
For \( j = 0 \rightarrow m \), \( L(0,j) = 0 \)

For \( i = 1 \rightarrow n \)
For \( j = 1 \rightarrow m \)
if \( X_i = Y_j \)
then \( L(i,j) = 1 + L(i-1,j-1) \)
else \( L(i,j) = \max \{ L(i-1,j), L(i,j-1) \} \)

Return \( L(n,m) \)
Running time:

For inputs of size $n$ & $m$, for loop of size $O(n)$ & inner loop of size $O(m)$

$\Rightarrow O(nm)$ total time.