Largest common subsequence (LCS):

Input: strings \( X = x_1x_2\ldots x_n \) \& \( Y = y_1\ldots y_n \)

Goal: length of longest string \( Z \) which is a subsequence in both \( X \) \& \( Y \).

Example:
\[
X = BCDBCDABA \\
Y = ABECBAA
\]

Solution = 4 corresponding to BCBA

Application: UNIX diff

Step 1: Define subproblem

Attempt 1: Consider prefixes.

For \( 0 \leq i \leq n \),
\[
L(i) = \text{length of LCS in } x_1, \ldots, x_i \text{ \& } y_1, \ldots, y_i
\]
Can we express $L(i)$ in terms of $L(1), \ldots, L(i-1)$?

Earlier example: BCDBCDA
ABECBA

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

So take best of $L(6)$

But what if $X_i \neq Y_i$? E.g., $L(6)$.

Might use $X_i$ with $Y_j$ for $j < i$

or $Y_i$ with $X_j$

then left with different length strings.

Attempt 2: For $0 \leq ij \leq n$,

let $L(i,j) =$ length of LCS of $X_1, \ldots, X_i$
with $Y_1, \ldots, Y_j$

(can think of $n$ replaced by $n \& n$)

length of $X$ length of $Y$
2-dimensional table

Base cases: 1st row & 1st column.
L(i,0) = 0 & L(0,j) = 0

For i > 1, j > 1, let's figure out how to express L(i,j) in terms of smaller

Two cases to consider:
1) \( X_i = Y_j \) or 2) \( X_i \neq Y_j \)

Case 1: \( X_i = Y_j \):
If LCS ends with \( X_i = Y_j \) then: \( L(i,j) = 1 + L(i-1,j-1) \)
What if \( X_i \neq Y_j \) not used then can match together & get longer CS.
If \( X_i \) matched with \( Y_k \) for \( k < j \) then can match with \( Y_j \) instead to get same length.

So, \( L(i,j) = 1 + L(i-1,j-1) \).
Suppose $X_i = Y_j$:

Either $X_i$ not matched or $Y_j$ not matched (can't both be matched)

Hence, $L(i,j) = \max\{L(i-1,j), L(i,j-1)\}$

Summarizing:

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

How to fill the table?

Row by row (or column by column)

$$L = \begin{array}{c}
\text{Row 1} \\
\vdots \\
\text{Row 3} \\
\end{array}$$
LCS(\(X, Y\)): \(X = x_1 \ldots x_n, Y = y_1 \ldots y_m\)

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

Return \(L(n, m)\).

Running time:
For inputs of size \(n \& m\),
one for loop of size \(O(n)\)
2 inner loop of \(O(m)\)
\(\Rightarrow O(nm)\) total time.