Dynamic Programming:
Accelerate the evaluation of recursive funs

Thm:

\[ Z = \text{an arbitrary LCS of } x \text{ and } y \]
\[ k = \text{the length of } Z \]

\[ \text{If } x[n] = y[m] \Rightarrow \]
\[ 2[k] = x[n] = y[m] \]
\[ Z[1:] = k-1 \]

\[ = \text{an LCS of } x[1:n-1] \text{ and } y[1:m-1] \]

Input:
\[ x: \text{a string of length } n \]
\[ y: \text{a string of length } m \]

Subsequence
\[ s \text{ is a subsequence of } t \]
\[ \text{if at least one of the following holds} \]
\[ \cdot s = t \text{ or } \]
\[ \cdot \text{ we can convert } t \text{ to } s \text{ by deleting chars} \]

\[ t = \text{ABCDEF} \]
\[ s = \text{BD} \]
\[ s = \text{BA} \]
\[ s = \emptyset \]

\[ \text{LCS length of } x \text{ and } y \]
\[ \text{LCS of the trimmed } x \text{ and } y \]

Goal: Find a common subsequence of \( x \) and \( y \) with the maximum length

\[ x = \text{ABCBDADB} \]
\[ y = \text{BDCABA} \]

\[ \text{LCS: } \text{BCBA}, \quad \text{BCAB} \]

\[ x = \emptyset \]
\[ y = \text{BDLABA} \]

\[ \text{LCS: } \emptyset \]

\[ \text{If } x[n] \neq y[m] \Rightarrow \]
\[ \text{at least one of the following correct} \]
\[ \cdot Z = \text{an LCS of } \]
\[ x[1:n-1] \text{ and } y \]
\[ \cdot Z = \text{an LCS of } \]
\[ x \text{ and } y[1:m-1] \]

\[ x = \text{ABCBDADB} \]
\[ y = \text{BDCABA} \]

\[ Z = \text{BCAB} \]

\[ x = \text{ABCDEF} \]
\[ y = \text{BDCABA} \]

\[ Z = \text{BCAB} \]