Longest Common Subsequence Problem

In the longest-common-subsequence problem, we are given two sequences \( X = \langle x_1, x_2, \ldots, x_m \rangle \) and \( Y = \langle y_1, y_2, \ldots, y_n \rangle \) and wish to find a maximum-length common subsequence of \( X \) and \( Y \).

- \( axbyce \) LCS is \( abc \)
- \( aibjce \) LCS is \( abc \)
- \( wxbyce \) LCS is \( abc \)
- \( zaybxcw \) LCS is \( yc \)

Dynamic Programming Solution

- LCS is \( amio \)
- LCS is \( otin \)
- LCS is \( moin \)
if first characters match,
then find LCS of rest of strings
of
$S[2...m]$ and $T[2...n]$.

One string is $S[1...m]$
other string is $T[1...n]$

Try the following three possibilities

$S[1...m]$  $S[2...m]$  $S[2...m]$
$T[2...n]$  $T[1...n]$  $T[2...n]$

End up wanting LCS of $S[i...m]$ and $T[j...n]$

Write a recursive alg $LCS(S,i,T,j)$
\[ \text{LCS}(S, i, T, j) \]

- if \( i > S.\text{length} \) or \( j > T.\text{length} \)
  - return 0
- if \( S[i] = T[j] \)
  - return \( 1 + \text{LCS}(S, i+1, T, j+1) \)
- else
  - \( \text{val1} = \text{LCS}(S, i, T, j+1) \)
  - \( \text{val2} = \text{LCS}(S, i+1, T, j) \)
  - \( \text{val3} = \text{LCS}(S, i+1, T, j+1) \)
  - return \( \max(\text{val1}, \text{val2}, \text{val3}) \)

\[ \text{Table}[i, j] = 1 + \text{LCS}(S, i+1, T, j+1) \]
- return \( \text{Table}[i, j] \)

\[ \text{Table}[i, j] = \max(\text{val1}, \text{val2}, \text{val3}) \]
- return \( \text{Table}[i, j] \)