Longest Common Subsequence

- \( C = c_1 \ldots c_g \) is a subsequence of \( A = a_1 \ldots a_m \) if \( C \) can be obtained by removing elements from \( A \) (but retaining order)
- LCS(\( A, B \)): A maximum length sequence that is a subsequence of both \( A \) and \( B \)

\[
\begin{align*}
\text{occuranec} & \quad \text{attaggct} \\
\text{occurrence} & \quad \text{tacgacca}
\end{align*}
\]

Determine the LCS of the following strings

BARTHOLEMEWSIMPSON

KRUSTYTHECLOWN

String Alignment Problem

- Align sequences with gaps
- Charge \( \delta_x \) if character \( x \) is unmatched
- Charge \( \gamma_{xy} \) if character \( x \) is matched to character \( y \)

Note: the problem is often expressed as a minimization problem, with \( \gamma_{xx} = 0 \) and \( \delta_x > 0 \)

LCS Optimization

- \( A = a_1 a_2 \ldots a_m \)
- \( B = b_1 b_2 \ldots b_n \)
- \( \text{Opt}[j,k] \) is the length of LCS(\( a_1 a_2 \ldots a_j, b_1 b_2 \ldots b_k \))

Optimization recurrence

If \( a_j = b_k \):
\[
\text{Opt}[j,k] = 1 + \text{Opt}[j-1,k-1]
\]

If \( a_j \neq b_k \):
\[
\text{Opt}[j,k] = \max(\text{Opt}[j-1,k], \text{Opt}[j,k-1])
\]
Give the Optimization Recurrence for the String Alignment Problem

- Charge $\delta_x$ if character $x$ is unmatched
- Charge $\gamma_{xy}$ if character $x$ is matched to character $y$

$Opt[j, k] =$

Let $a_j = x$ and $b_k = y$
Express as minimization

Dynamic Programming Computation

Code to compute $Opt[j,k]$

A[1..m], B[1..n]
for $i := 1$ to $m$  $Opt[i, 0] := 0$;
for $j := 1$ to $n$  $Opt[0,j] := 0$;
$Opt[0,0] := 0$;
for $i := 1$ to $m$
  for $j := 1$ to $n$
    if $A[i] = B[j]$  
      $Opt[i,j] := 1 + Opt[i-1,j-1]$;  
      $Best[i,j] := Diag$;
    else if $Opt[i-1,j] \geq Opt[i,j-1]$
      $Opt[i,j] := Opt[i-1,j], Best[i,j] := Left$;
    else
      $Opt[i,j] := Opt[i,j-1], Best[i,j] := Down$;

Storing the path information

Observations about the Algorithm

- The computation can be done in $O(m+n)$ space if we only need one column of the $Opt$ values or Best Values
- The algorithm can be run from either end of the strings

How good is this algorithm?

- Is it feasible to compute the LCS of two strings of length 100,000 on a standard desktop PC? Why or why not.