Dynamic Programming

Four-step (two-phase) method:

1. Characterize the structure of an optimal solution
2. Recursively define the value of an optimal solution
3. Compute the value of an optimal solution in a bottom-up fashion
4. Construct an optimal solution from computed information
Problem statement:
Problem statement:

**Input:** Sequences

\[
X_m = \langle x_1, x_2, x_3, \ldots, x_m \rangle \\
Y_n = \langle y_1, y_2, \ldots, y_n \rangle
\]
Longest Common Subsequence (LCS) – DP case study 3

Problem statement:

**Input: Sequences**

\[ X_m = \langle x_1, x_2, x_3, \ldots, x_m \rangle \]
\[ Y_n = \langle y_1, y_2, \ldots, y_n \rangle \]

**Output:** longest common subsequence (LCS) of \( X_m \) and \( Y_n \)
LCS

Terminology

1. Sequence, e.g.
   - $X_7 = \langle A, B, C, B, D, A, B \rangle$
   - ALGORITHM
LCS

Terminology

1. Sequence, e.g.
   - $X_7 = \langle A, B, C, B, D, A, B \rangle$
   - ALGORITHM

2. Subsequence, e.g.
   - $\langle A, C, D, B \rangle$ is a subsequence of $X$
   - ART is a subsequence ALGORITHM
LCS

Terminology

1. Sequence, e.g.
   - $X_7 = \langle A, B, C, B, D, A, B \rangle$
   - ALGORITHM

2. Subsequence, e.g.
   - $\langle A, C, D, B \rangle$ is a subsequence of $X$
   - ART is a subsequence ALGORITHM

3. Common subsequence, e.g.
   - Given $X_7 = \langle A, B, C, B, D, A, B \rangle$
     $Y_6 = \langle B, D, C, A, B, A \rangle$
   - $Z_3 = \langle B, C, A \rangle$ is a common subsequence of $X_7$ and $Y_6$
   - $Z_4 = \langle B, C, B, A \rangle$ is also a common subsequence of $X_7$ and $Y_6$

4. Longest common subsequence (LCS), e.g.
   - $Z_4$ is a longest common subsequence (LCS) of $X_7$ and $Y_6$
   - LCS is not unique, $\langle B, C, A, B \rangle$ is also a LCS.
LCS

Terminology

1. **Sequence**, e.g.
   - \(X_7 = \langle A, B, C, B, D, A, B \rangle\)
   - ALGORITHM

2. **Subsequence**, e.g.
   - \(\langle A, C, D, B \rangle\) is a subsequence of \(X\)
   - ART is a subsequence ALGORITHM

3. **Common subsequence**, e.g.
   - Given \(X_7 = \langle A, B, C, B, D, A, B \rangle\)
     \(Y_6 = \langle B, D, C, A, B, A \rangle\)
   - \(Z_3 = \langle B, C, A \rangle\) is a common subsequence of \(X_7\) and \(Y_6\)
   - \(Z_4 = \langle B, C, B, A \rangle\) is also a common subsequence of \(X_7\) and \(Y_6\)

4. **Longest common subsequence (LCS)**, e.g.
   - \(Z_4\) is a longest common subsequence (LCS) of \(X_7\) and \(Y_6\)
   - LCS is not unique, \(\langle B, C, A, B \rangle\) is also a LCS.
LCS

A brute-force solution:

▶ For every subsequence of $X_m$, check if it is a subsequence of $Y_n$. 

$\text{Running time: } \Theta(n \cdot 2^m)$

Intractable!
A brute-force solution:

- For every subsequence of $X_m$, check if it is a subsequence of $Y_n$.

- Running time: $\Theta(n \cdot 2^m)$

- Intractable!
LCS

DP – step 1: characterize the structure of an optimal solution
DP – step 1: *characterize the structure of an optimal solution*

Let $Z_k = \langle z_1, z_2, \ldots, z_k \rangle$ be any LCS of

$$X_m = \langle x_1, x_2, \ldots, x_m \rangle \quad \text{and} \quad Y_n = \langle y_1, \ldots, y_n \rangle$$

Then
DP – step 1: characterize the structure of an optimal solution

Let $Z_k = \langle z_1, z_2, \ldots, z_k \rangle$ be any LCS of

\[ X_m = \langle x_1, x_2, \ldots, x_m \rangle \quad \text{and} \quad Y_n = \langle y_1, \ldots, y_n \rangle \]

Then

- **Case 1.** If $x_m = y_n$, then
  
  (a) $z_k = x_m = y_n$
  
  (b) $Z_{k-1} = \langle z_1, z_2, \ldots, z_{k-1} \rangle = \text{LCS}(X_{m-1}, Y_{n-1})$
LCS

DP – step 1: *characterize the structure of an optimal solution*

Let \( Z_k = \langle z_1, z_2, \ldots, z_k \rangle \) be any LCS of

\[
X_m = \langle x_1, x_2, \ldots, x_m \rangle \quad \text{and} \quad Y_n = \langle y_1, \ldots, y_n \rangle
\]

Then

- **Case 1.** If \( x_m = y_n \), then
  
  (a) \( z_k = x_m = y_n \)
  
  (b) \( Z_{k-1} = \langle z_1, z_2, \ldots, z_{k-1} \rangle = \text{LCS}(X_{m-1}, Y_{n-1}) \)

- **Case 2.** If \( x_m \neq y_n \), then
  
  (a) \( z_k \neq x_m \implies Z_k = \text{LCS}(X_{m-1}, Y_n) \)
  
  (b) \( z_k \neq y_n \implies Z_k = \text{LCS}(X_m, Y_{n-1}) \)
LCS

DP – step 1: characterize the structure of an optimal solution

Let \( Z_k = \langle z_1, z_2, \ldots, z_k \rangle \) be any LCS of

\[
X_m = \langle x_1, x_2, \ldots, x_m \rangle \quad \text{and} \quad Y_n = \langle y_1, \ldots, y_n \rangle
\]

Then

- **Case 1.** If \( x_m = y_n \), then
  
  \[
  \begin{align*}
  (a) & \quad z_k = x_m = y_n \\
  (b) & \quad Z_{k-1} = \langle z_1, z_2, \ldots, z_{k-1} \rangle = \text{LCS}(X_{m-1}, Y_{n-1})
  \end{align*}
  \]

- **Case 2.** If \( x_m \neq y_n \), then
  
  \[
  \begin{align*}
  (a) & \quad z_k \neq x_m \implies Z_k = \text{LCS}(X_{m-1}, Y_n) \\
  (b) & \quad z_k \neq y_n \implies Z_k = \text{LCS}(X_m, Y_{n-1})
  \end{align*}
  \]

In words, the optimal solution to the (whole) problem contains within it the optimal solutions to subproblems.
LCS

DP – step 1: characterize the structure of an optimal solution

Let $Z_k = \langle z_1, z_2, \ldots, z_k \rangle$ be any LCS of

$$X_m = \langle x_1, x_2, \ldots, x_m \rangle \quad \text{and} \quad Y_n = \langle y_1, \ldots, y_n \rangle$$

Then

- **Case 1.** If $x_m = y_n$, then
  
  (a) $z_k = x_m = y_n$
  
  (b) $Z_{k-1} = \langle z_1, z_2, \ldots, z_{k-1} \rangle = \text{LCS}(X_{m-1}, Y_{n-1})$

- **Case 2.** If $x_m \neq y_n$, then
  
  (a) $z_k \neq x_m \implies Z_k = \text{LCS}(X_{m-1}, Y_n)$
  
  (b) $z_k \neq y_n \implies Z_k = \text{LCS}(X_m, Y_{n-1})$

In words, the optimal solution to the (whole) problem contains within it the optimal solutions to subproblems = **the optimal substructure property**
LCS

DP – step 2: *recursively define the value of an optimal solution*
LCS

DP – step 2: recursively define the value of an optimal solution

Define

\[ c[i, j] = \text{length of LCS}(X_i, Y_j) \]
LCS

DP – step 2: *recursively define the value of an optimal solution*

- Define

  \[ c[i, j] = \text{length of LCS}(X_i, Y_j) \]

- \[ c[m, n] = \text{length of LCS}(X_m, Y_n) \]

- For initialization:

  \[ c[i, 0] = c[0, j] = 0 \]

- By Case 1 of the optimal structure property:
  - If \( x_i = y_j \), then
    \[ Z_{\ell} = \{ x_i, y_j \} = \text{LCS}(X_{i-1}, Y_{j-1}) \]
    \[ c[i, j] = c[i-1, j-1] + 1 \]

- By Case 2 of the optimal structure property:
  - If \( x_i \neq y_j \), then
    \[ Z_{\ell} = \begin{cases} \text{LCS}(X_{i-1}, Y_j) & \text{if } x_i \neq y_j \\ \text{LCS}(X_i, Y_{j-1}) & \text{if } x_i = y_j \end{cases} \]
    \[ c[i, j] = \max\{c[i-1, j], c[i, j-1]\} \]
LCS

DP – step 2: recursively define the value of an optimal solution

▸ Define

\[ c[i, j] = \text{length of LCS}(X_i, Y_j) \]

▸ \[ c[m, n] = \text{length of LCS}(X_m, Y_n) \]

▸ \[ c[i, 0] = c[0, j] = 0 \] for initialization
LCS

**DP – step 2: recursively define the value of an optimal solution**

- Define
  
  \[ c[i, j] = \text{length of LCS}(X_i, Y_j) \]

- \( c[m, n] = \text{length of LCS}(X_m, Y_n) \)

- \( c[i, 0] = c[0, j] = 0 \) for initialization

- By **Case 1** of the optimal structure property: if \( x_i = y_j \), then
  
  \[(a)\] \( z_\ell = x_i = y_j \)
  
  \[(b)\] \( Z_{\ell-1} = \langle z_1, z_2, \ldots, z_{\ell-1} \rangle = \text{LCS}(X_{i-1}, Y_{j-1}) \)
LCS

DP – step 2: recursively define the value of an optimal solution

- Define

\[ c[i, j] = \text{length of LCS}(X_i, Y_j) \]

- \( c[m, n] = \text{length of LCS}(X_m, Y_n) \)

- \( c[i, 0] = c[0, j] = 0 \) for initialization

- By Case 1 of the optimal structure property: if \( x_i = y_j \), then

\[
\begin{align*}
(a) & \quad z_\ell = x_i = y_j \\
(b) & \quad Z_{\ell-1} = \langle z_1, z_2, \ldots, z_{\ell-1} \rangle = \text{LCS}(X_{i-1}, Y_{j-1})
\end{align*}
\]

we have

\[ c[i, j] = c[i - 1, j - 1] + 1 \]
LCS

DP – step 2: recursively define the value of an optimal solution

▸ Define

\[ c[i, j] = \text{length of LCS}(X_i, Y_j) \]

▸ \( c[m, n] = \text{length of LCS}(X_m, Y_n) \)

▸ \( c[i, 0] = c[0, j] = 0 \) for initialization

▸ By Case 1 of the optimal structure property: if \( x_i = y_j \), then

(a) \( z_\ell = x_i = y_j \)

(b) \( Z_{\ell-1} = \langle z_1, z_2, \ldots, z_{\ell-1} \rangle = \text{LCS}(X_{i-1}, Y_{j-1}) \)

we have

\[ c[i, j] = c[i - 1, j - 1] + 1 \]

▸ By Case 2 of the optimal structure property: if \( x_i \neq y_j \), then

(a) \( z_\ell \neq x_i \implies Z_\ell = \text{LCS}(X_{i-1}, Y_j) \)

(b) \( z_\ell \neq y_j \implies Z_\ell = \text{LCS}(X_i, Y_{j-1}) \)
LCS

DP – step 2: \textit{recursively define the value of an optimal solution}

\begin{itemize}
\item Define
\[
    c[i, j] = \text{length of LCS}(X_i, Y_j)
\]
\item \(c[m, n] = \text{length of LCS}(X_m, Y_n)\)
\item \(c[i, 0] = c[0, j] = 0\) for initialization
\item By Case 1 of the optimal structure property: if \(x_i = y_j\), then
\[
\begin{align*}
    (a) & \quad z_\ell = x_i = y_j \\
    (b) & \quad Z_{\ell-1} = \langle z_1, z_2, \ldots, z_{\ell-1} \rangle = \text{LCS}(X_{i-1}, Y_{j-1})
\end{align*}
\]
we have
\[
    c[i, j] = c[i - 1, j - 1] + 1
\]
\item By Case 2 of the optimal structure property: if \(x_i \neq y_j\), then
\[
\begin{align*}
    (a) & \quad z_\ell \neq x_i \implies Z_\ell = \text{LCS}(X_{i-1}, Y_j) \\
    (b) & \quad z_\ell \neq y_j \implies Z_\ell = \text{LCS}(X_i, Y_{j-1})
\end{align*}
\]
we have
\[
    c[i, j] = \max\{c[i, j - 1], c[i - 1, j]\}
\]
\end{itemize}
In summary,

\[
    c[i, j] = \begin{cases} 
        0 & \text{if } i = 0 \text{ or } j = 0 \text{ (initials)} \\
        c[i - 1, j - 1] + 1 & \text{if } x[i] = y[j] \text{ (Case 1)} \\
        \max\{c[i, j - 1], c[i - 1, j]\} & \text{if } x[i] \neq y[j] \text{ (Case 2)}
    \end{cases}
\]
LCS

- In summary,

\[
c[i, j] = \begin{cases} 
0 & \text{if } i = 0 \text{ or } j = 0 \text{ (initials)} \\
  c[i - 1, j - 1] + 1 & \text{if } x[i] = y[j] \quad \text{(Case 1)} \\
  \max\{c[i, j - 1], c[i - 1, j]\} & \text{if } x[i] \neq y[j] \quad \text{(Case 2)} 
\end{cases}
\]

- Meanwhile, create \( b[i, j] \) to record the optimal subproblem solution chosen when computing \( c[i, j] \)
LCS

DP – step 3: compute \( c[i, j] \) (and \( b[i, j] \)) in a bottom-up approach

- Compute \( c[i, j] \) and \( b[i, j] \) in a bottom-up approach.
  - \( c[i, j] \) is the length of \( \text{LCS}(X_i, Y_j) \)
  - \( b[i, j] \) shows how to construct the corresponding \( \text{LCS}(X_i, Y_j) \)
LCS

DP – step 3: *compute* $c[i, j]$ (*and* $b[i, j]$) *in a bottom-up approach*

- Compute $c[i, j]$ and $b[i, j]$ in a **bottom-up approach**.
  - $c[i, j]$ is the length of LCS($X_i, Y_j$)
  - $b[i, j]$ shows how to construct the corresponding LCS($X_i, Y_j$)

- **Cost:**
  - Running time: $\Theta(mn)$
  - Space: $\Theta(mn)$
LCS

LCS-length(X,Y)
set c[i,0] = 0 and c[0,j] = 0
for i = 1 to m // Row-major order to compute c and b arrays
  for j = 1 to n
    if X(i) = Y(j)
      c[i,j] = c[i-1,j-1] + 1
      b[i,j] = 'Diag' // go to up diagonal
    elseif c[i-1,j] >= c[i,j-1]
      c[i,j] = c[i-1,j]
      b[i,j] = 'Up' // go up
    else
      c[i,j] = c[i,j-1]
      b[i,j] = 'Left' // go left
    endif
  endfor
endfor
return c and b
LCS

DP – step 4: *construct an optimal solution from computed information*
Example: \(X_7 = \langle A, B, C, B, D, A, B \rangle\) and \(Y_6 = \langle B, D, C, A, B, A \rangle\)
Example: \( X_7 = \langle A, B, C, B, D, A, B \rangle \) and \( Y_6 = \langle B, D, C, A, B, A \rangle \)

\[
c[\cdot, \cdot] + b[\cdot, \cdot]
\]

<table>
<thead>
<tr>
<th>( i )</th>
<th>( y_j )</th>
<th>( B )</th>
<th>( D )</th>
<th>( C )</th>
<th>( A )</th>
<th>( B )</th>
<th>( A )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>( x_i )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>B</td>
<td>0</td>
<td>1</td>
<td>←1</td>
<td>1</td>
<td>1</td>
<td>←2</td>
</tr>
<tr>
<td>3</td>
<td>C</td>
<td>0</td>
<td>↑</td>
<td>↑</td>
<td>↑</td>
<td>↑</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>B</td>
<td>0</td>
<td>1</td>
<td>↑</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>D</td>
<td>0</td>
<td>↑</td>
<td>←1</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>A</td>
<td>0</td>
<td>↑</td>
<td>↑</td>
<td>↑</td>
<td>↑</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>B</td>
<td>0</td>
<td>↑</td>
<td>↑</td>
<td>↑</td>
<td>↑</td>
<td>4</td>
</tr>
</tbody>
</table>

15.4 Longest common subsequence 395

![Table and Diagram](image-url)
**Example:** $X_7 = \langle A, B, C, B, D, A, B \rangle$ and $Y_6 = \langle B, D, C, A, B, A \rangle$

$c[\cdot, \cdot] + b[\cdot, \cdot]$

\[\begin{array}{ccccccc}
   & & & & & & \\
   & & & & & & \\
   i & j & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\
   0 & x_i & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
   1 & A & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
   2 & B & 0 & 1 & 1 & 1 & 2 & 2 & 2 \\
   3 & C & 0 & 1 & 1 & 2 & 2 & 2 & 2 \\
   4 & B & 0 & 1 & 1 & 2 & 2 & 3 & 3 \\
   5 & D & 0 & 1 & 2 & 2 & 3 & 3 & 3 \\
   6 & A & 0 & 1 & 2 & 2 & 3 & 3 & 4 \\
   7 & B & 0 & 1 & 2 & 2 & 3 & 4 & 4 \\
\end{array}\]

(1) Length of LCS = $c[7, 6] = 4$

(2) By the $b$-table ("\(\uparrow\), \(\leftarrow\), \(\swarrow\)"), the LCS is $B C B A$