Dynamic Programming: Longest Subsequence

Ira A. Fulton Schools of Engineering
Arizona State University

DP Part 1

- **The LCS problem and naïve solution**
- **Principles of Dynamic Programming**
  - Optimal substructure
  - Recursion
  - Avoiding solving the same instance multiple times
  - Construct an optimal solution
- **Example and Summary**

Longest Common Subsequence

- Problem: Given two sequences X and Y, Compute a longest common subsequence (LCS) of X and Y.

- Example:

X=<A,B,C,B,D,A,B>, Y=<B,D,C,A,B,A>.

Z=<B,C,B,A> is a longest common subsequence of X and Y

- Z is a subsequence of X=<A,B,C,B,D,A,B>.
- Z is a subsequence of Y=<B,D,C,A,B,A>.
- Hence Z is a common subsequence of X and Y.
- Is it the longest?

Longest Common Subsequence

- What is an LCS of X and Y?
- 1st, it should be a common subsequence of X and Y.
- 2nd, it should have the maximum length among all common subsequences of X and Y.
- We need to define related concepts precisely.

Subsequence

- Let X=< $x_1, x_2, \ldots, x_m$ > and Z=< $z_1, z_2, \ldots, z_k$ >.
- Z is said to be a subsequence of X if there exists a strictly increasing sequence < $i_1, i_2, \ldots, i_k$ > of indices of X such that
- $\mathrm{z_j} = x_{i_j}$ for $j = 1, 2, \ldots, k$.

- Example: Z=<B,C,B,A> is a subsequence of X=<A,B,C,B,D,A,B>.
- The index sequence is <2,3,4,6>.
- <B> is a subsequence of X.
- The index sequence can be <2>, <4> or <7>

Common Subsequence

- Given sequences X and Y, a sequence Z is a common subsequence of X and Y if Z is a subsequence of X and a subsequence of Y.

- Example:

X=<A,B,C,B,D,A,B>, Y=<B,D,C,A,B,A>.

Examples for common subsequences of X and Y are:

<>

<A>, <B>, <C>, <D>

<A,B>, <B,A>, <B,C>, <B,D>

<B,B,A>, <B,C,A>, <B,C,B>

<B,C,B,A>

Length of a sequence

- Given a sequence $X = <x_1, x_2, \ldots, x_m>$, the length of X is m, which is the number of symbols in X.
- The length of an empty sequence <> is 0
- The length of the sequence X=<a,b,c> is 3
- We denote the length of sequence X by |X|

Prefix

- Let X=<x_1, x_2, …, x_m> be a sequence.
- We define **the i-th prefix of X**, for i=0, 1, 2, …, m, as
- X_i=<x_1, x_2, …, x_i>

Longest Common Subsequence (LCS)

Let X and Y be two sequences. A sequence Z is a longest common subsequence (LCS) of X and Y, if

(1) Z is a common subsequence of X and Y;

(2) Z is longest among all common subsequences of X and Y.

Computing LCS: naive approach

```
Exhaustive-LCS(X, Y)
Find the set of symbols in X and Y
Generate S: all sequences with length≤min(|X|, |Y|)
Sorted S by decreasing length
for all s∈S do
    if (s is a common subsequence of X and Y){
        return s
    }
```

Bad News: Exponential running time!

Ira A. Fulton Schools of Engineering

Arizona State University

DP Part 2

The LCS problem and naïve solution

Principles of Dynamic Programming

- Optimal substructure
- Recursion
- Avoiding solving the same instance multiple times
- Construct an optimal solution

Example and Summary

Steps in Dynamic Programming

- **Characterize the structure of an optimal solution**
  - Most important part of DP
- **Define the value of an optimal solution**
  - May lead to exponential running time
- **Compute the value of an optimal solution bottom-up**
  - Do not solve the same instance multiple times
- **Construct an optimal solution from computed information**
- **We will use an example to illustrate these steps.**

LCS: Structure of an optimal solution

Theorem 14.1: Optimal Substructure of An LCS

Let $X = \langle x_1, x_2, \ldots, x_m \rangle$ and $Y = \langle y_1, y_2, \ldots, y_n \rangle$ be two sequences.

Let $Z = \langle z_1, z_2, \ldots, z_k \rangle$ be any LCS of X and Y.

(1) If $x_m = y_n$, then $z_k = x_m = y_n$ and Z_{k-1} is an LCS of X_{m-1} and Y_{n-1}

(2) If $x_m \neq y_n$, and $z_k \neq x_m$, then Z is an LCS of X_{m-1} and Y

(3) If $x_m \neq y_n$, and $z_k \neq y_n$, then Z is an LCS of X and Y_{n-1}

LCS: Structure of an optimal solution

Theorem 14.1: Optimal Substructure of An LCS

Let $X = <x_1, x_2, \ldots, x_m>$ and $Y = <y_1, y_2, \ldots, y_n>$ be two sequences.

Let **$Z = <z_1, z_2, \ldots, z_k>$ be any LCS of X and Y.**

(1) If $x_m = y_n$, then $z_k = x_m = y_n$ and $Z_{k\text{-}1}$ is an LCS of $X_{m\text{-}1}$ and $Y_{n\text{-}1}$

(2) If $x_m \neq y_n$, and $z_k \neq x_m$, then Z is an LCS of $X_{m\text{-}1}$ and Y

(3) If $x_m \neq y_n$, and $z_k \neq y_n$, then Z is an LCS of X and $Y_{n\text{-}1}$

Proof. Case 1.

If $z_k \neq x_m$ then we could append x_m to Z to obtain a common subsequence of X and Y with length k+1. This is a contradiction.

Therefore $z_k = x_m = y_n$. Now $Z_{k\text{-}1}$ is a common subsequence of $X_{m\text{-}1}$ and $Y_{n\text{-}1}$ with length k-1. We need to prove that $Z_{k\text{-}1}$ is an LCS of $X_{m\text{-}1}$ and $Y_{n\text{-}1}$. If not, Z will have length larger than k, another contradiction. Therefore Case 1 is proved.

LCS: Structure of an optimal solution

Theorem 14.1: Optimal Substructure of An LCS

Let $X = <x_1, x_2, \ldots, x_m>$ and $Y = <y_1, y_2, \ldots, y_n>$ be two sequences.

***Let* $Z = <z_1, z_2, \ldots, z_k>$ be any LCS of X and Y.**

(1) **If $x_m = y_n$, then $z_k = x_m = y_n$ and Z_{k-1} is an LCS of X_{m-1} and Y_{n-1}**

(2) **If $x_m \neq y_n$, and $z_k \neq x_m$, then Z is an LCS of X_{m-1} and Y**

(3) **If $x_m \neq y_n$, and $z_k \neq y_n$, then Z is an LCS of X and Y_{n-1}**

Proof. Case 2.

If $z_k \neq x_m$ then Z is a common subsequence of X_{m-1} and Y.

If Z is not an LCS of X_{m-1} and Y, we will have a longer common subsequence of X_{m-1} and Y, which is also a common subsequence of X and Y, a contradiction. This proves Case 2.

LCS: Structure of an optimal solution

Theorem 14.1: Optimal Substructure of An LCS

Let $X = <x_1, x_2, \ldots, x_m>$ and $Y = <y_1, y_2, \ldots, y_n>$ be two sequences.

Let **$Z = <z_1, z_2, \ldots, z_k>$ be any LCS of X and Y.**

(1) If $x_m = y_n$, then $z_k = x_m = y_n$ and Z_{k-1} is an LCS of X_{m-1} and Y_{n-1}

(2) If $x_m \neq y_n$, and $z_k \neq x_m$, then Z is an LCS of X_{m-1} and Y

(3) If $x_m \neq y_n$, and $z_k \neq y_n$, then Z is an LCS of X and Y_{n-1}

Proof. Case 3.

Similar to Case 2.

Examples

(1) If $x_m = y_n$, then $z_k = x_m = y_n$ and Z_{k-1} is an LCS of X_{m-1} and Y_{n-1}

(2) If $x_m \neq y_n$, and $z_k \neq x_m$, then Z is an LCS of X_{m-1} and Y

(3) If $x_m \neq y_n$, and $z_k \neq y_n$, then Z is an LCS of X and Y_{n-1}

Example

X = <A, B, C, B, D, A>, Y = < B, D, C, A, B, A>

Z = <B, C, B, A>

Examples

(1) **If $x_m = y_n$, then $z_k = x_m = y_n$ and $Z_{k\text{-}1}$ is an LCS of $X_{m\text{-}1}$ and $Y_{n\text{-}1}$**

(2) **If $x_m \neq y_n$, and $z_k \neq x_m$, then Z is an LCS of $X_{m\text{-}1}$ and Y**

(3) **If $x_m \neq y_n$, and $z_k \neq y_n$, then Z is an LCS of X and $Y_{n\text{-}1}$**

Example

X = <A, B, C, B, D, A>, Y = < B, D, C, A, B, A>

Z = <B, C, B, A>

Examples

(1) If $x_m = y_n$, then $z_k = x_m = y_n$ and $Z_{k\text{-}1}$ is an LCS of $X_{m\text{-}1}$ and $Y_{n\text{-}1}$

(2) If $x_m \neq y_n$, and $z_k \neq x_m$, then Z is an LCS of $X_{m\text{-}1}$ and Y

(3) If $x_m \neq y_n$, and $z_k \neq y_n$, then Z is an LCS of X and $Y_{n\text{-}1}$

Example

X = <A, B, C, B, D, A>, Y = < B, D, C, A, B, A>

Z = <B, C, B, A>

Examples

(1) **If $x_m = y_n$, then $z_k = x_m = y_n$ and Z_{k-1} is an LCS of X_{m-1} and Y_{n-1}**

(2) **If $x_m \neq y_n$, and $z_k \neq x_m$, then Z is an LCS of X_{m-1} and Y**

(3) **If $x_m \neq y_n$, and $z_k \neq y_n$, then Z is an LCS of X and Y_{n-1}**

Example

X = <A, B, C, B, D, *A*>, Y = < B, D, C, A, B, *A*>

Z = <B, C, B, *A*>

Examples

(1) **If $x_m = y_n$, then $z_k = x_m = y_n$ and $Z_{k\text{-}1}$ is an LCS of $X_{m\text{-}1}$ and $Y_{n\text{-}1}$**

(2) **If $x_m \neq y_n$, and $z_k \neq x_m$, then Z is an LCS of $X_{m\text{-}1}$ and Y**

(3) **If $x_m \neq y_n$, and $z_k \neq y_n$, then Z is an LCS of X and $Y_{n\text{-}1}$**

Example

X = <A, B, C, B, D, *A*>, Y = < B, D, C, A, B, *A*>

Z = <B, C, B, *A*>

Examples

(1) **If $x_m = y_n$, then $z_k = x_m = y_n$ and $Z_{k\text{-}1}$ is an LCS of $X_{m\text{-}1}$ and $Y_{n\text{-}1}$**

(2) **If $x_m \neq y_n$, and $z_k \neq x_m$, then Z is an LCS of $X_{m\text{-}1}$ and Y**

(3) **If $x_m \neq y_n$, and $z_k \neq y_n$, then Z is an LCS of X and $Y_{n\text{-}1}$**

Example

X = <A, B, C, B, D, *A*>, Y = < B, D, C, A, B, *A*>

Z = <B, C, B, *A*>

Examples

(1) **If $x_m = y_n$, then $z_k = x_m = y_n$ and Z_{k-1} is an LCS of X_{m-1} and Y_{n-1}**

(2) **If $x_m \neq y_n$, and $z_k \neq x_m$, then Z is an LCS of X_{m-1} and Y**

(3) **If $x_m \neq y_n$, and $z_k \neq y_n$, then Z is an LCS of X and Y_{n-1}**

Example

X = <A, B, C, B, D, A>, Y = < B, D, C, A, B, A>

Z = <B, C, B, A>

Examples

(1) If $x_m = y_n$, then $z_k = x_m = y_n$ and Z_{k-1} is an LCS of X_{m-1} and Y_{n-1}

(2) If $x_m \neq y_n$, and $z_k \neq x_m$, then Z is an LCS of X_{m-1} and Y

(3) If $x_m \neq y_n$, and $z_k \neq y_n$, then Z is an LCS of X and Y_{n-1}

Example

X = <A, B, C, B, *D*, *A*>, Y = < B, D, C, A, B, *A*>

Z = <B, C, B, *A*>

Examples

(1) **If $x_m = y_n$, then $z_k = x_m = y_n$ and $Z_{k\text{-}1}$ is an LCS of $X_{m\text{-}1}$ and $Y_{n\text{-}1}$**

(2) **If $x_m \neq y_n$, and $z_k \neq x_m$, then Z is an LCS of $X_{m\text{-}1}$ and Y**

(3) **If $x_m \neq y_n$, and $z_k \neq y_n$, then Z is an LCS of X and $Y_{n\text{-}1}$**

Example

X = <A, B, C, B, *D*, *A*>, Y = < B, D, C, A, B, *A*>

Z = <B, C, B, *A*>

Examples

(1) **If $x_m = y_n$, then $z_k = x_m = y_n$ and Z_{k-1} is an LCS of X_{m-1} and Y_{n-1}**

(2) **If $x_m \neq y_n$, and $z_k \neq x_m$, then Z is an LCS of X_{m-1} and Y**

(3) **If $x_m \neq y_n$, and $z_k \neq y_n$, then Z is an LCS of X and Y_{n-1}**

Example

X = <A, B, C, *B*, *D*, *A*>, Y = < B, D, C, A, *B*, *A*>

Z = <B, C, *B*, *A*>

Examples

(1) If $x_m = y_n$, then $z_k = x_m = y_n$ and Z_{k-1} is an LCS of X_{m-1} and Y_{n-1}

(2) If $x_m \neq y_n$, and $z_k \neq x_m$, then Z is an LCS of X_{m-1} and Y

(3) If $x_m \neq y_n$, and $z_k \neq y_n$, then Z is an LCS of X and Y_{n-1}

Example

X = <A, B, C, *B*, *D*, *A*>, Y = < B, D, C, A, *B*, *A*>

Z = <B, C, *B*, *A*>

Examples

(1) **If $x_m = y_n$, then $z_k = x_m = y_n$ and $Z_{k\text{-}1}$ is an LCS of $X_{m\text{-}1}$ and $Y_{n\text{-}1}$**

(2) **If $x_m \neq y_n$, and $z_k \neq x_m$, then Z is an LCS of $X_{m\text{-}1}$ and Y**

(3) **If $x_m \neq y_n$, and $z_k \neq y_n$, then Z is an LCS of X and $Y_{n\text{-}1}$**

Example

X = <A, B, C, *B*, *D*, *A*>, Y = < B, D, C, A, *B*, *A*>

Z = <B, C, *B*, *A*>

Examples

(1) **If $x_m = y_n$, then $z_k = x_m = y_n$ and $Z_{k\text{-}1}$ is an LCS of $X_{m\text{-}1}$ and $Y_{n\text{-}1}$**

(2) **If $x_m \neq y_n$, and $z_k \neq x_m$, then Z is an LCS of $X_{m\text{-}1}$ and Y**

(3) **If $x_m \neq y_n$, and $z_k \neq y_n$, then Z is an LCS of X and $Y_{n\text{-}1}$**

Example

X = <A, B, C, B, D, A>, Y = < B, D, C, A, B, A>

Z = <B, C, B, A>

Examples

(1) If $x_m = y_n$, then $z_k = x_m = y_n$ and $Z_{k\text{-}1}$ is an LCS of $X_{m\text{-}1}$ and $Y_{n\text{-}1}$

(2) If $x_m \neq y_n$, and $z_k \neq x_m$, then Z is an LCS of $X_{m\text{-}1}$ and Y

(3) If $x_m \neq y_n$, and $z_k \neq y_n$, then Z is an LCS of X and $Y_{n\text{-}1}$

Example

X = <A, B, C, *B*, *D*, *A*>, Y = < B, D, C, *A*, *B*, *A*>

Z = <B, C, *B*, *A*>

Examples

(1) **If $x_m = y_n$, then $z_k = x_m = y_n$ and Z_{k-1} is an LCS of X_{m-1} and Y_{n-1}**

(2) **If $x_m \neq y_n$, and $z_k \neq x_m$, then Z is an LCS of X_{m-1} and Y**

(3) **If $x_m \neq y_n$, and $z_k \neq y_n$, then Z is an LCS of X and Y_{n-1}**

Example

X = <A, B, *C*, *B*, *D*, *A*>, Y = < B, D, *C*, *A*, *B*, *A*>

Z = <B, *C*, *B*, *A*>

Examples

(1) If $x_m = y_n$, then $z_k = x_m = y_n$ and Z_{k-1} is an LCS of X_{m-1} and Y_{n-1}

(2) If $x_m \neq y_n$, and $z_k \neq x_m$, then Z is an LCS of X_{m-1} and Y

(3) If $x_m \neq y_n$, and $z_k \neq y_n$, then Z is an LCS of X and Y_{n-1}

Example

X = <A, B, C, B, D, A>, Y = < B, D, C, A, B, A>

Z = <B, C, B, A>

Examples

(1) **If $x_m = y_n$, then $z_k = x_m = y_n$ and Z_{k-1} is an LCS of X_{m-1} and Y_{n-1}**

(2) **If $x_m \neq y_n$, and $z_k \neq x_m$, then Z is an LCS of X_{m-1} and Y**

(3) **If $x_m \neq y_n$, and $z_k \neq y_n$, then Z is an LCS of X and Y_{n-1}**

Example

X = <A, B, *C*, *B*, *D*, *A*>, Y = < B, D, *C*, *A*, *B*, *A*>

Z = <B, *C*, *B*, *A*>

Examples

(1) **If $x_m = y_n$, then $z_k = x_m = y_n$ and $Z_{k\text{-}1}$ is an LCS of $X_{m\text{-}1}$ and $Y_{n\text{-}1}$**

(2) **If $x_m \neq y_n$, and $z_k \neq x_m$, then Z is an LCS of $X_{m\text{-}1}$ and Y**

(3) **If $x_m \neq y_n$, and $z_k \neq y_n$, then Z is an LCS of X and $Y_{n\text{-}1}$**

Example

X = <A, B, *C*, *B*, *D*, *A*>, Y = < B, *D*, *C*, *A*, *B*, *A*>

Z = <B, *C*, *B*, *A*>

Examples

(1) If $x_m = y_n$, then $z_k = x_m = y_n$ and $Z_{k\text{-}1}$ is an LCS of $X_{m\text{-}1}$ and $Y_{n\text{-}1}$

(2) If $x_m \neq y_n$, and $z_k \neq x_m$, then Z is an LCS of $X_{m\text{-}1}$ and Y

(3) If $x_m \neq y_n$, and $z_k \neq y_n$, then Z is an LCS of X and $Y_{n\text{-}1}$

Example

X = <A, B, C, B, D, A>, Y = < B, D, C, A, B, A>

Z = <B, C, B, A>

Examples

(1) **If $x_m = y_n$, then $z_k = x_m = y_n$ and $Z_{k\text{-}1}$ is an LCS of $X_{m\text{-}1}$ and $Y_{n\text{-}1}$**

(2) **If $x_m \neq y_n$, and $z_k \neq x_m$, then Z is an LCS of $X_{m\text{-}1}$ and Y**

(3) **If $x_m \neq y_n$, and $z_k \neq y_n$, then Z is an LCS of X and $Y_{n\text{-}1}$**

Example

X = <A, *B*, *C*, *B*, *D*, *A*>, Y = < *B*, *D*, *C*, *A*, *B*, *A*>

Z = <*B*, *C*, *B*, *A*>

Examples

(1) If $x_m = y_n$, then $z_k = x_m = y_n$ and $Z_{k\text{-}1}$ is an LCS of $X_{m\text{-}1}$ and $Y_{n\text{-}1}$

(2) If $x_m \neq y_n$, and $z_k \neq x_m$, then Z is an LCS of $X_{m\text{-}1}$ and Y

(3) If $x_m \neq y_n$, and $z_k \neq y_n$, then Z is an LCS of X and $Y_{n\text{-}1}$

Example

X = <A, B, C, B, D, A>, Y = < B, D, C, A, B, A>

Z = <B, C, B, A>

Examples

(1) If $x_m = y_n$, then $z_k = x_m = y_n$ and Z_{k-1} is an LCS of X_{m-1} and Y_{n-1}

(2) If $x_m \neq y_n$, and $z_k \neq x_m$, then Z is an LCS of X_{m-1} and Y

(3) If $x_m \neq y_n$, and $z_k \neq y_n$, then Z is an LCS of X and Y_{n-1}

Example

X = <A, B, C, B, D, A>, Y = < B, D, C, A, B, A>

Z = <B, C, B, A>

Examples

(1) **If $x_m = y_n$, then $z_k = x_m = y_n$ and Z_{k-1} is an LCS of X_{m-1} and Y_{n-1}**

(2) **If $x_m \neq y_n$, and $z_k \neq x_m$, then Z is an LCS of X_{m-1} and Y**

(3) **If $x_m \neq y_n$, and $z_k \neq y_n$, then Z is an LCS of X and Y_{n-1}**

Example

X = <*A*, *B*, *C*, *B*, *D*, *A*>, Y = < *B*, *D*, *C*, *A*, *B*, *A*>

Z = <*B*, *C*, *B*, *A*>

Observations from the optimal substructure

- Observation: Finding an LCS of X and Y can be reduce to finding

(1) LCS X_{m-1} and Y_{n-1}

(2) LCS X and Y_{n-1}

(3) LCS X_{m-1} and Y

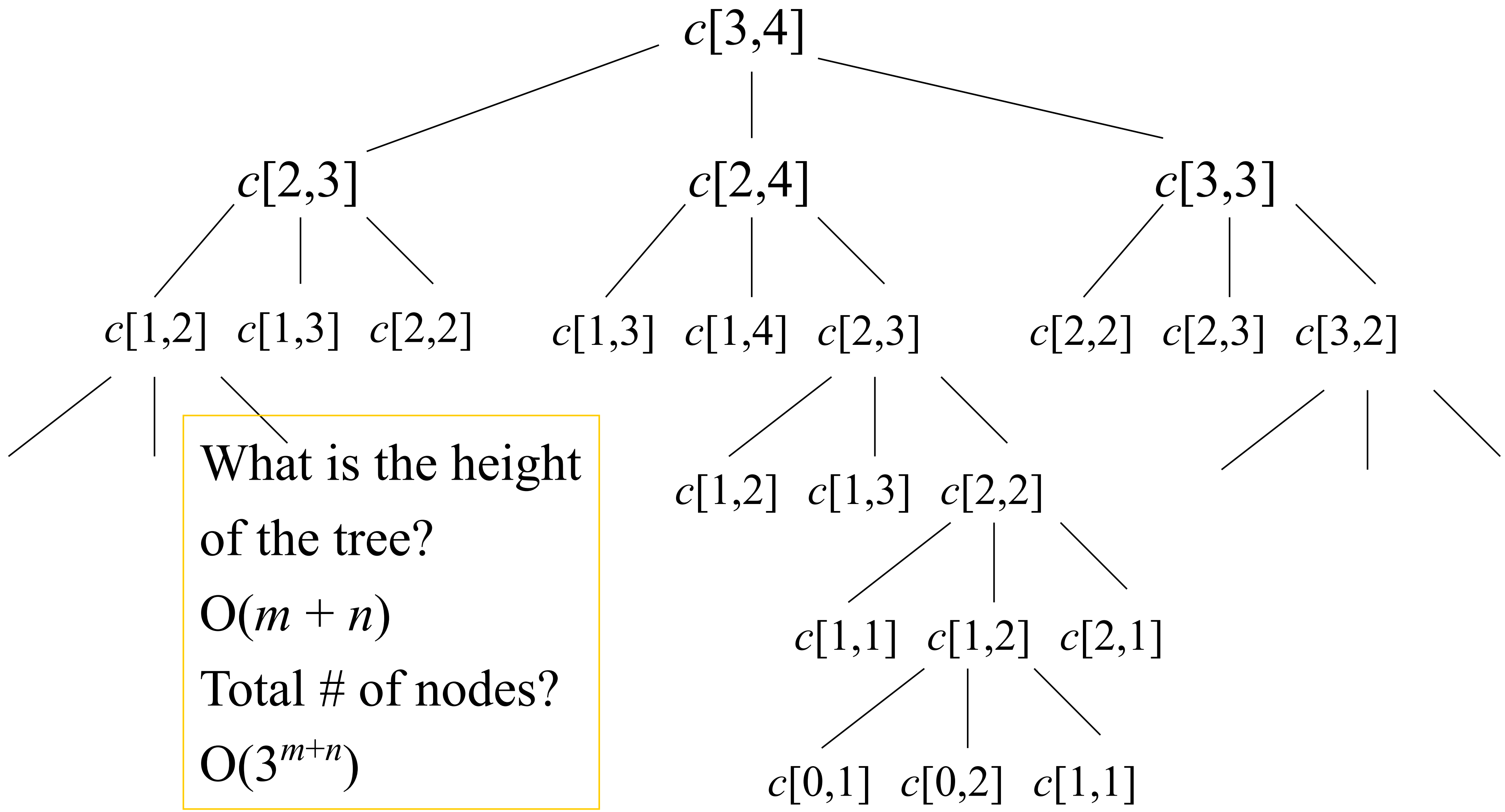
- Each of these are smaller instances of the same problem.

Define the value of an optimal solution

- Define the value: Let *c[i, j]* be the length of an LCS of X_i and Y_j.
- Obviously, if $i = 0$ or $j = 0$, then $c[i, j] = 0$.
- In general, we have

$$c[i,j] = \begin{cases} 0 & i = 0 \quad or \quad j = 0 \\ c[i-1, j-1] + 1 & i, j > 0 \quad and \quad x_i = y_j \\ \max(c[i, j-1], c[i-1, j]) & i, j > 0 \quad and \quad x_i \neq y_j \end{cases}$$

Computing the optimal values top-down



What is the height of the tree?

$O(m + n)$

Total # of nodes?

$O(3^{m+n})$

Computing the optimal values bottom-up

```
LCS-Length(X_m, Y_n)
for i := 0 to m do c[i, 0] := 0
for j := 1 to n do c[0, j] := 0
for i := 1 to m do
   for j := 1 to n do
      if x_i == y_j
         c[i, j] := c[i-1, j-1] + 1
         b[i, j] := "↖"
      else if c[i-1, j] ≥ c[i, j-1]
         c[i, j] := c[i-1, j]
         b[i, j] := "↑"
      else
         c[i, j] := c[i, j-1]
         b[i, j] := "←"
return matrices c and b
```

Time complexity: $O(mn)$

Construct an optimal solution

```
Print-LCS(b, X, i, j)
if i == 0 or j == 0
  return
if b[i, j] == "↖"
  Print-LCS(b, X, i-1, j-1)
  print x_i
else if b[i, j] == "↑"
  Print-LCS(b, X, i-1, j)
else
  Print-LCS(b, X, i, j-1)
```

Time complexity: $O(m+n)$

Ira A. Fulton Schools of
Engineering
Arizona State University

DP Part 3

The LCS problem and naïve solution

Principles of Dynamic Programming

- Optimal substructure
- Recursion
- Avoiding solving the same instance multiple times
- Construct an optimal solution

Example and Summary

Example: Computing optimal values

i	$c[i, j]$ x_i	j = 0, $y_j =$	1, **B**	2, *D*	3, **C**	4, *A*	5, **B**	6, **A**
0		0	0	0	0	0	0	0
1	*A*	0	↑ 0	↑ 0	↑ 0	↖ 1	← 1	↖ 1
2	**B**	0	↖ 1	← 1	← 1	↑ 1	↖ 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	↑ 2	↑ 3	↑ 3
6	**A**	0	↑ 1	↑ 2	↑ 2	↖ 3	↑ 3	↖ 4
7	*B*	0	↖ 1	↑ 2	↑ 2	↑ 3	↖ 4	↑ 4

Example: Computing optimal values

$c[i, j]$

i		j	0	1	2	3	4	5	6
	x_i	$y_j =$		**B**	D	**C**	A	**B**	**A**
0			0	0	0	0	0	0	0
1	A		0	↑ 0	↑ 0	↑ 0	↖ 1	← 1	↖ 1
2	**B**		0	↖ 1	← 1	← 1	↑ 1	↖ 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	↑ 2	↑ 3	↑ 3
6	**A**		0	↑ 1	↑ 2	↑ 2	↖ 3	↑ 3	↖ 4
7	B		0	↖ 1	↑ 2	↑ 2	↑ 3	↖ 4	↑ 4

Example: Computing optimal values

$c[i, j]$

i	j	0	1	2	3	4	5	6
		$y_j =$	**B**	*D*	**C**	*A*	**B**	**A**
0	x_i	0	0	0	0	0	0	0
1	|| *A*	0	↑ 0	↑ 0	↑ 0	↖ 1	← 1	↖ 1
2	**B**	0	↖ 1	← 1	← 1	↑ 1	↖ 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	↑ 2	↑ 3	↑ 3
6	**A**	0	↑ 1	↑ 2	↑ 2	↖ 3	↑ 3	↖ 4
7	*B*	0	↖ 1	↑ 2	↑ 2	↑ 3	↖ 4	↑ 4

Example: Computing optimal values

j		0	1	2	3	4	5	6
i	$c[i, j]$	y_j =	**B**	*D*	**C**	*A*	**B**	**A**
0	x_i	0	0	0	0	0	0	0
1	|| *A*	0	↑ 0	↑ 0	↑ 0	↖ 1	← 1	↖ 1
2	**B**	0	↖ 1	← 1	← 1	↑ 1	↖ 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	↑ 2	↑ 3	↑ 3
6	**A**	0	↑ 1	↑ 2	↑ 2	↖ 3	↑ 3	↖ 4
7	*B*	0	↖ 1	↑ 2	↑ 2	↑ 3	↖ 4	↑ 4

Example: Computing optimal values

$c[i, j]$

j		0	1	2	3	4	5	6
i		$y_j =$	**B**	*D*	**C**	*A*	**B**	**A**
0	x_i	0	0	0	0	0	0	0
1	|| *A*	0	↑ 0	↑ 0	↑ 0	↖ 1	← 1	↖ 1
2	**B**	0	↖ 1	← 1	← 1	↑ 1	↖ 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	↑ 2	↑ 3	↑ 3
6	**A**	0	↑ 1	↑ 2	↑ 2	↖ 3	↑ 3	↖ 4
7	*B*	0	↖ 1	↑ 2	↑ 2	↑ 3	↖ 4	↑ 4

Example: Computing optimal values

i	$c[i, j]$	j = 0	1	2	3	4	5	6
	x_i	$y_j =$	**B**	*D*	**C**	*A*	**B**	**A**
0		0	0	0	0	0	0	0
1	*A*	0	↑ 0	↑ 0	↑ 0	↖ 1	← 1	↖ 1
2	**B**	0	↖ 1	← 1	← 1	↑ 1	↖ 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	↑ 2	↑ 3	↑ 3
6	**A**	0	↑ 1	↑ 2	↑ 2	↖ 3	↑ 3	↖ 4
7	*B*	0	↖ 1	↑ 2	↑ 2	↑ 3	↖ 4	↑ 4

Example: Computing optimal values

	j	0	1	2	3	4	5	6
i	$c[i, j]$	$y_j =$	**B**	*D*	**C**	*A*	**B**	**A**
0	x_i	0	0	0	0	0	0	0
1	$\parallel$ *A*	0	↑ 0	↑ 0	↑ 0	↖ 1	← 1	↖ 1
2	**B**	0	↖ 1	← 1	← 1	↑ 1	↖ 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	↑ 2	↑ 3	↑ 3
6	**A**	0	↑ 1	↑ 2	↑ 2	↖ 3	↑ 3	↖ 4
7	*B*	0	↖ 1	↑ 2	↑ 2	↑ 3	↖ 4	↑ 4

Example: Computing optimal values

i	x_i	j = 0	1	2	3	4	5	6
	$c[i,j]$	$y_j =$	B	D	C	A	B	A
0		0	0	0	0	0	0	0
1	A	0	↑ 0	↑ 0	↑ 0	↖ 1	← 1	↖ 1
2	B	0	↖ 1	← 1	← 1	↑ 1	↖ 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	↑ 2	↑ 3	↑ 3
6	A	0	↑ 1	↑ 2	↑ 2	↖ 3	↑ 3	↖ 4
7	B	0	↖ 1	↑ 2	↑ 2	↑ 3	↖ 4	↑ 4

Example: Computing optimal values

i	$c[i,j]$	j = 0	1	2	3	4	5	6
		$y_j =$	**B**	D	**C**	A	**B**	**A**
0	x_i	0	0	0	0	0	0	0
1	A	0	↑ 0	↑ 0	↑ 0	↖ 1	← 1	↖ 1
2	**B**	0	↖ 1	← 1	← 1	↑ 1	↖ 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	↑ 2	↑ 3	↑ 3
6	**A**	0	↑ 1	↑ 2	↑ 2	↖ 3	↑ 3	↖ 4
7	B	0	↖ 1	↑ 2	↑ 2	↑ 3	↖ 4	↑ 4

Example: Computing optimal values

j		0	1	2	3	4	5	6
i	$c[i, j]$	$y_j =$	**B**	D	**C**	A	**B**	**A**
0	x_i	0	0	0	0	0	0	0
1	A	0	↑ 0	↑ 0	↑ 0	↖ 1	← 1	↖ 1
2	**B**	0	↖ 1	← 1	← 1	↑ 1	↖ 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	↑ 2	↑ 3	↑ 3
6	**A**	0	↑ 1	↑ 2	↑ 2	↖ 3	↑ 3	↖ 4
7	B	0	↖ 1	↑ 2	↑ 2	↑ 3	↖ 4	↑ 4

Example: Computing optimal values

j		0	1	2	3	4	5	6
i	$c[i, j]$	$y_j =$	**B**	D	**C**	A	**B**	**A**
0	x_i	0	0	0	0	0	0	0
1	A	0	↑ 0	↑ 0	↑ 0	↖ 1	← 1	↖ 1
2	**B**	0	↖ 1	← 1	← 1	↑ 1	↖ 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	↑ 2	↑ 3	↑ 3
6	**A**	0	↑ 1	↑ 2	↑ 2	↖ 3	↑ 3	↖ 4
7	B	0	↖ 1	↑ 2	↑ 2	↑ 3	↖ 4	↑ 4

Example: Computing optimal values

j		0	1	2	3	4	5	6
i	$c[i, j]$	$y_j =$	**B**	D	**C**	A	**B**	**A**
0	x_i	0	0	0	0	0	0	0
1	$\|\|$ A	0	$\uparrow$ 0	$\uparrow$ 0	$\uparrow$ 0	$\nwarrow$ 1	$\leftarrow$ 1	$\nwarrow$ 1
2	**B**	0	$\nwarrow$ 1	$\leftarrow$ 1	$\leftarrow$ 1	$\uparrow$ 1	$\nwarrow$ 2	$\leftarrow$ 2
3	**C**	0	$\uparrow$ 1	$\uparrow$ 1	$\nwarrow$ 2	$\leftarrow$ 2	$\uparrow$ 2	$\uparrow$ 2
4	**B**	0	$\nwarrow$ 1	$\uparrow$ 1	$\uparrow$ 2	$\uparrow$ 2	$\nwarrow$ 3	$\leftarrow$ 3
5	D	0	$\uparrow$ 1	$\nwarrow$ 2	$\uparrow$ 2	$\uparrow$ 2	$\uparrow$ 3	$\uparrow$ 3
6	**A**	0	$\uparrow$ 1	$\uparrow$ 2	$\uparrow$ 2	$\nwarrow$ 3	$\uparrow$ 3	$\nwarrow$ 4
7	B	0	$\nwarrow$ 1	$\uparrow$ 2	$\uparrow$ 2	$\uparrow$ 3	$\nwarrow$ 4	$\uparrow$ 4

Example: Computing optimal values

$c[i, j]$

j		0	1	2	3	4	5	6
i	x_i	$y_j =$	**B**	D	**C**	A	**B**	**A**
0		0	0	0	0	0	0	0
1	A	0	↑ 0	↑ 0	↑ 0	↖ 1	← 1	↖ 1
2	**B**	0	↖ 1	← 1	← 1	↑ 1	↖ 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	↑ 2	↑ 3	↑ 3
6	**A**	0	↑ 1	↑ 2	↑ 2	↖ 3	↑ 3	↖ 4
7	B	0	↖ 1	↑ 2	↑ 2	↑ 3	↖ 4	↑ 4

Example: Computing optimal values

j		0	1	2	3	4	5	6
i	$c[i, j]$	$y_j =$	**B**	*D*	**C**	*A*	**B**	**A**
0	x_i	0	0	0	0	0	0	0
1	|| *A*	0	↑ 0	↑ 0	↑ 0	↖ 1	← 1	↖ 1
2	**B**	0	↖ 1	← 1	← 1	↑ 1	↖ 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	↑ 2	↑ 3	↑ 3
6	**A**	0	↑ 1	↑ 2	↑ 2	↖ 3	↑ 3	↖ 4
7	*B*	0	↖ 1	↑ 2	↑ 2	↑ 3	↖ 4	↑ 4

Example: Construct an optimal solution

i \ j		0	1	2	3	4	5	6
i	$c[i, j]$	$y_j =$	**B**	*D*	**C**	*A*	**B**	**A**
0	x_i =	0	0	0	0	0	0	0
1	*A*	0	↑ 0	↑ 0	↑ 0	↖ 1	← 1	↖ 1
2	**B**	0	↖ 1	← 1	← 1	↑ 1	↖ 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	↑ 2	↑ 3	↑ 3
6	**A**	0	↑ 1	↑ 2	↑ 2	↖ 3	↑ 3	↖ 4
7	*B*	0	↖ 1	↑ 2	↑ 2	↑ 3	↖ 4	↑ 4

Example: Construct an optimal solution

i	$c[i, j]$	j = 0	1	2	3	4	5	6
		y_j =	**B**	*D*	**C**	*A*	**B**	**A**
0	x_i	0	0	0	0	0	0	0
1	*A*	0	↑ 0	↑ 0	↑ 0	↖ 1	← 1	↖ 1
2	**B**	0	↖ 1	← 1	← 1	↑ 1	↖ 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	↑ 2	↑ 3	↑ 3
6	**A**	0	↑ 1	↑ 2	↑ 2	↖ 3	↑ 3	↖ 4
7	*B*	0	↖ 1	↑ 2	↑ 2	↑ 3	↖ 4	↑ 4

Example: Construct an optimal solution

i	$c[i,j]$							
	j	0	1	2	3	4	5	6
	x_i ||	$y_j =$	**B**	*D*	**C**	*A*	**B**	**A**
0		0	0	0	0	0	0	0
1	*A*	0	↑ 0	↑ 0	↑ 0	↖ 1	← 1	↖ 1
2	**B**	0	↖ 1	← 1	← 1	↑ 1	↖ 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	↑ 2	↑ 3	↑ 3
6	**A**	0	↑ 1	↑ 2	↑ 2	↖ 3	↑ 3	↖ 4
7	*B*	0	↖ 1	↑ 2	↑ 2	↑ 3	↖ 4	↑ 4

Example: Construct an optimal solution

j		0	1	2	3	4	5	6
i	$c[i, j]$	$y_j =$	**B**	*D*	**C**	*A*	**B**	**A**
0	x_i	0	0	0	0	0	0	0
1	‖ *A*	0	↑ 0	↑ 0	↑ 0	↖ 1	← 1	↖ 1
2	**B**	0	↖ 1	← 1	← 1	↑ 1	↖ 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	↑ 2	↑ 3	↑ 3
6	**A**	0	↑ 1	↑ 2	↑ 2	↖ 3	↑ 3	↖ 4
7	*B*	0	↖ 1	↑ 2	↑ 2	↑ 3	↖ 4	↑ 4

Example: Construct an optimal solution

j		0	1	2	3	4	5	6
i	$c[i, j]$	$y_j =$	**B**	D	**C**	A	**B**	**A**
0	$x_i =$	0	0	0	0	0	0	0
1	A	0	↑ 0	↑ 0	↑ 0	↖ 1	← 1	↖ 1
2	**B**	0	↖ 1	← 1	← 1	↑ 1	↖ 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	↑ 2	↑ 3	↑ 3
6	**A**	0	↑ 1	↑ 2	↑ 2	↖ 3	↑ 3	↖ 4
7	B	0	↖ 1	↑ 2	↑ 2	↑ 3	↖ 4	↑ 4

Example: Construct an optimal solution

	j	0	1	2	3	4	5	6
i	$c[i, j]$	$y_j =$	**B**	*D*	**C**	*A*	**B**	**A**
0	x_i =	0	0	0	0	0	0	0
1	*A*	0	↑ 0	↑ 0	↑ 0	↖ 1	← 1	↖ 1
2	**B**	0	↖ 1	← 1	← 1	↑ 1	↖ 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	↑ 2	↑ 3	↑ 3
6	**A**	0	↑ 1	↑ 2	↑ 2	↖ 3	↑ 3	↖ 4
7	*B*	0	↖ 1	↑ 2	↑ 2	↑ 3	↖ 4	↑ 4

Example: Construct an optimal solution

	j	0	1	2	3	4	5	6
i	$c[i, j]$	$y_j =$	**B**	D	**C**	A	**B**	**A**
0	x_i	0	0	0	0	0	0	0
1	|| A	0	↑ 0	↑ 0	↑ 0	↖ 1	← 1	↖ 1
2	**B**	0	↖ 1	← 1	← 1	↑ 1	↖ 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	↑ 2	↑ 3	↑ 3
6	**A**	0	↑ 1	↑ 2	↑ 2	↖ 3	↑ 3	↖ 4
7	B	0	↖ 1	↑ 2	↑ 2	↑ 3	↖ 4	↑ 4

Example: Construct an optimal solution

i	$c[i, j]$ / j	0	1	2	3	4	5	6
	x_i || $y_j =$		**B**	*D*	**C**	*A*	**B**	**A**
0		0	0	0	0	0	0	0
1	*A*	0	↑ 0	↑ 0	↑ 0	↖ 1	← 1	↖ 1
2	**B**	0	↖ 1	← 1	← 1	↑ 1	↖ 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	↑ 2	↑ 3	↑ 3
6	**A**	0	↑ 1	↑ 2	↑ 2	↖ 3	↑ 3	↖ 4
7	*B*	0	↖ 1	↑ 2	↑ 2	↑ 3	↖ 4	↑ 4

Example: Construct an optimal solution

j		0	1	2	3	4	5	6
i	$c[i, j]$	$y_j =$	**B**	*D*	**C**	*A*	**B**	**A**
0	x_i	0	0	0	0	0	0	0
1	|| *A*	0	↑ 0	↑ 0	↑ 0	↖ 1	← 1	↖ 1
2	**B**	0	↖ 1	← 1	← 1	↑ 1	↖ 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	↑ 2	↑ 3	↑ 3
6	**A**	0	↑ 1	↑ 2	↑ 2	↖ 3	↑ 3	↖ 4
7	*B*	0	↖ 1	↑ 2	↑ 2	↑ 3	↖ 4	↑ 4

More Examples?

- Given a sequence of n integers. Find its longest monotonically increasing subsequence.

- An $O(n^2)$ time algorithm.

DP Algorithm for Computing Fibonacci Numbers?

- Declare array F[1:n].
- F[1] := 1
- F[2] := 1
- For i=3 to n do
  - F[i] := F[i-1] + F[i-2]
- EndFor

- What is the time complexity?

- $O(n)$

Summary

- Dynamic programming is an optimal algorithm based on optimal structures of sub-problems.
- We fill up a table to avoid solving the same sub-problem more than once.
- As another example, we have solved the LCS problem.
- Please practice this to make sure you fully understand it.

Ira A. Fulton Schools of Engineering

Arizona State University