G54 AAD

L 11 LCS & DYNAMIC PROGRAMMING

Definition

Given a sequence $X = (x_1, \dots, x_m)$,
a sequence $Z = (z_1, \dots, z_k)$,
is called a <u>subsequence</u> of X if
there exists a strictly increasing sequence $\langle i_1, \dots, i_k \rangle$
such that $z_j = x_{i_j}$ for all $j \in [k]$

<u>Ex</u>

	1	2	3	4	5	6	7	8	9
X =	o	o	a	o	b	o	c	o	o
Z =	a	b	c						

$\Rightarrow Z$ is a subsequence of X.

$i_1 = 3$

$i_2 = 5$

$i_3 = 7$

Definition

Given two sequences X and Y,

a sequence Z is called a

common subsequence of X and Y if

- Z is a subsequence of X, and
- Z is a subsequence of Y.

Example

$X = 0 \quad 1 \quad 2 \quad a \quad 1 \quad 2 \quad b \quad 0 \quad c \quad 1$

$Y = 0 \quad a \quad 0 \quad 0 \quad b \quad c \quad 1$

$Z = a \quad b \quad c$

LONGEST COMMON SUBSEQUENCE (LCS)

Given two input sequences X and Y, the LCS problem consists in finding the longest common subsequence of X and Y.

Example

X = 0 0 0 0 a 0 0 0 b 0 0 0 c

Y = 0 a 0 0 b 0 0 c

Z = 0 a 0 0 b 0 0 c

NAIVE SOLUTION (BRUTE FORCE)

Z = ϵ (empty string)
for each subsequence S of X
  if S is a subsequence of Y and
  S is longer than Z,
  then
    Z = S
return Z

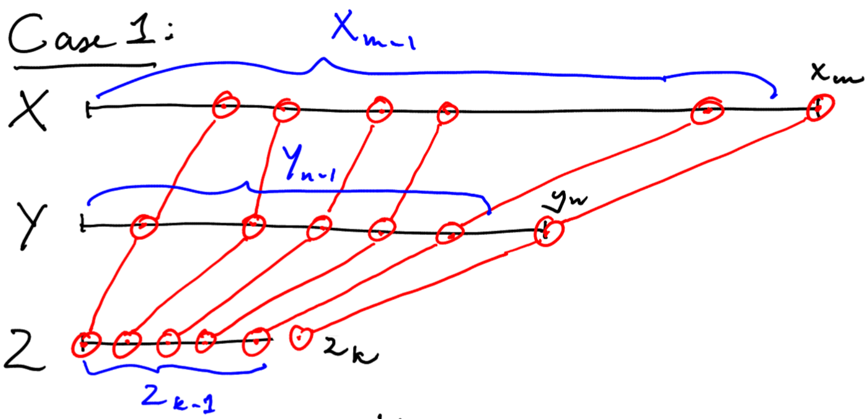
- The number of subsequences of X is exponentially large in the length of X.

⇒ Brute force is inefficient on LCS.

Theorem

Given two seqences $X = (x_1, x_2, \ldots, x_m)$,
$Y = (y_1, y_2, \ldots, y_n)$,
let $Z = (z_1, \ldots, z_k)$ be any LCS of X and Y.

Case 1:



Notation
Given $X = (x_1, \ldots, x_m)$
then for $i < m$
$X_i = (x_1, \ldots, x_i)$

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

Proof (by contradiction)

If $z_k \neq x_m$, we could make Z longer by appending x_m (which equals y_n). This would contradict that Z is the longest common subsequence. Hence, $z_k = x_m = y_n$.

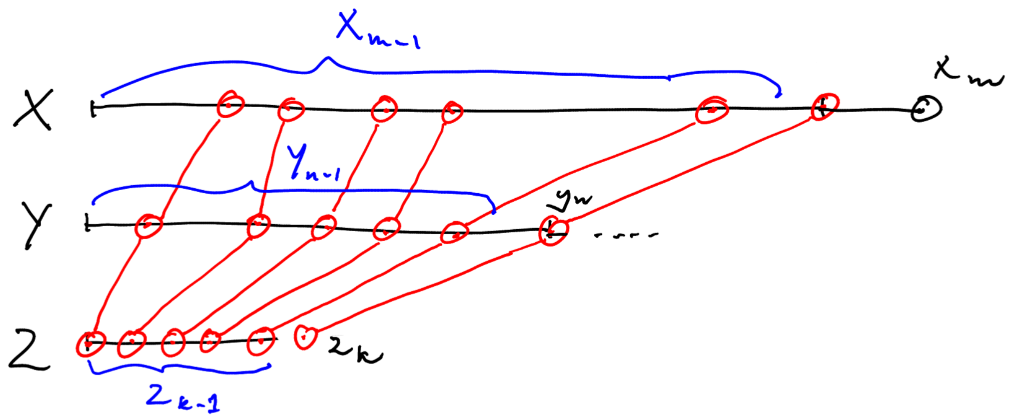
Z_{k-1} is a common subsequence of X_{m-1} and Y_{n-1}, and it has length $k-1$.

If Z_{k-1} was not the longest common subsequence of X_{m-1} and Y_{n-1}, then there must have been a different common subsequence U of X_{m-1} and Y_{n-1} with length at least k.
However, the sequence Uz_k would then be an LCS of X and Y of length $k+1$, which contradicts that Z is an LCS.

Case 2:

If $x_m \neq y_n$ and $x_m \neq z_k$ then

Z is the LCS of X_{m-1} and Y.

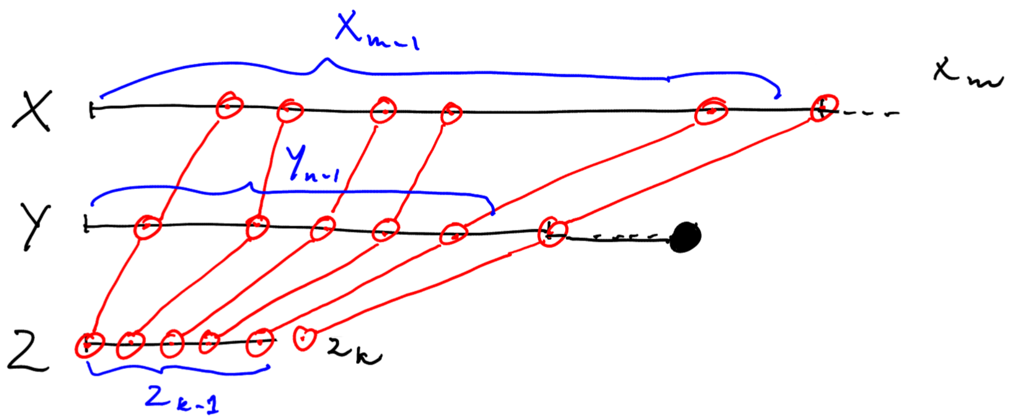


Proof

If $z_k \neq x_m$, then Z is a common subsequence of length k of X_{m-1} and Y. If there were a different common subsequence W of length $k+1$, it would also be a subsequence of X and Y. Hence, Z must be a LCS of X_{m-1} and Y.

Case 2b:

If $x_m \neq y_n$ and $x_m \neq z_k$ then Z is the LCS of X_{m-1} and Y.



Proof

Symmetric to case 2a.

Theorem (Optimal Substructure of an LCS)

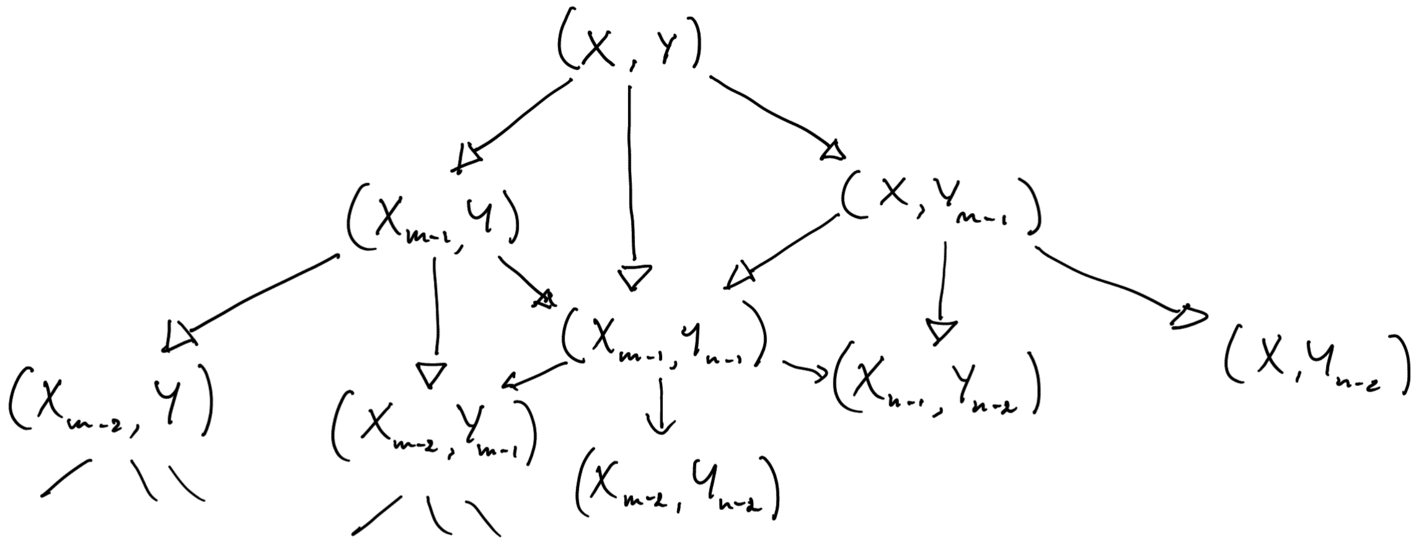
Let $X = (x_1, \ldots, x_m)$ and $Y = (y_1, \ldots, y_n)$ be two sequences, and $Z = (z_1, \ldots, z_k)$ an LCS of X and Y.

- 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}
- If $x_m \neq y_n$ and $x_m \neq z_k$ then Z is an LCS of X_{m-1} and Y
- If $x_m \neq y_n$ and $y_n \neq z_k$ then Z is an LCS of X and Y_{n-1}.

$\Rightarrow$ If I know the LCS of X and Y_{n-1}, X_{m-1} and Y, and X_{m-1} and Y_{n-1}, then I can compute the LCS of X and Y.

DIVIDE & CONQUER

Split the problem into sub-problems,
solve the sub-problems recursively,
combine the solutions to the subproblems.



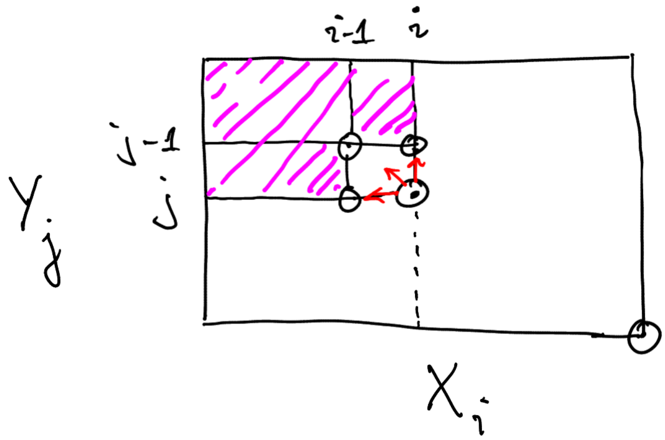
$\Rightarrow$ The problems have overlapping substructure!
We recompute previous solutions.

$\Rightarrow$ Divide & Conquer wastes computations!

DYNAMIC PROGRAMMING

Divide the problem into sub-problems, Solve each sub-problem only once by saving solutions in a table.

Length of LCS



$$c[i,j] = \begin{cases} 0 & \text{if } i=0 \text{ or } j=0 \\ c[i-1,j-1]+1 & \text{if } x_i = y_j \\ \max\{c[i-1,j], c[i,j-1]\} & \text{otherwise} \end{cases}$$

$c[i,j]$: length of LCS of X_i and Y_j