DNA and Text sequence Alignment

- **Subsequence:**

Given a string $X = x_0 x_1 x_2 \ldots x_{n-1}$

A subsequence of X is a string S such that:

(1) $S = x_{i_1} x_{i_2} \ldots x_{i_K}$

where $i_j < i_{j+1}$

- not necessarily contiguous
- but must be in order given by X.

- **Example Subsequences.**

$X = CGATAATTGAGA$

$S_1 = CGAT$

$S_2 = ATTT$

$S_3 = CATT$

Not subsequence: ACGA
CGCA

The longest common subsequence (LCS) problem

Problem description:

Given 2 strings:

$$X = x_0 x_1 x_2 \ldots x_{n-1}$$

$$Y = y_0 y_1 y_2 \ldots y_{m-1}$$

Find the longest ~~sub~~string S such that:

S is a subsequence of X
S is a subsequence of Y

Example:

$X = ABCABCABC$

$Y = BABACBAB$

Common subsequence:

AAA → A B C A B C A B C
AAA → B A B A C B A B

ABAB → A B C A B C A B C
ABAB → B A B A C B A B

ABABA → A B C A B C A B C
ABABA → B A B A C B A B

What is the longest sequence?

Naive Solution: brute force.

(1) enumerate all subsequences of X

(2) Take the longest one that is a subsequence of Y.

Problem: exponential number of subsequence of X:

$$X = x_0 x_1 x_2 x_3 \ldots x_{n-1}$$

↓

select or not.

$$2 \cdot 2 \cdot 2 \cdot \ldots \cdot 2$$

2^n possible sub sequence!

To test this many will take ages !!!

Eg: $2^{64} = 1844 \ldots$

takes 584 years to finish
if one sequence take 10^{-9} sec!

Dynamic Programming (recursion)

If: problem(n) is decomposable into a number of smaller problems:

problem(n_1), problem(n_2), ...

with: $n_1 < n$, $n_2 < n$, ...

And: The solution of problem(n_1), problem(n_2), ... can be used to solve

problem(n)

EASILY

Then we can apply "Dynamic Programming" (recursion) to solve the problem.

Examples:

Fib(n) = Fib(n-1) + Fib(n-2)

Hanoi(n) = Hanoi(n-1) followed by another Hanoi(n-1)

A dynamic Programming Solution to LCS

- Define: $X[0..i] = x_0 x_1 x_2 \dots x_i$
  $Y[0..j] = y_0 y_1 y_2 \dots y_j$

- Define:

  $L[i,j]$ = length of the longest string that is a subsequence of $X[0..i] = x_0 x_1 \dots x_i$ and $Y[0..j] = y_0 y_1 \dots y_j$.

- We want to find $L[n,m]$.

- Define $L[i,j]$ as smaller problems (LCS)
  ⇓
  Decompose.

$X[0..i] = x_0 x_1 \dots x_{i-1} x_i$

$Y[0..j] = y_0 y_1 \dots y_{j-1} y_j$

→ longest subseq = $L[i,j]$

(1) $x_i = y_j$:

match.

$x_0 x_1 \dots x_{i-1} \; x_i$

$y_0 y_1 \dots y_{j-1} \; y_j$

longest subsequence in here + x_i y_j

longest subseq.'s length = $L[i-1, j-1]$.

$\Rightarrow L[i,j] = L[i-1, j-1] + 1.$

(2) $x_i \neq y_j$:

$x_0 x_1 \dots x_{i-1} \; x_i$ ≠ different.

$y_0 y_1 \dots y_{j-1} \; y_j$

Then we must find $L[i,j]$ matching letters in:

$\begin{cases} x_0 x_1 \dots x_{i-1} \\ y_0 y_1 \dots y_{j-1} \end{cases}$ or $\begin{cases} x_0 x_1 \dots x_{i-1} \, x_i \\ y_0 y_1 \dots y_{j-1} \end{cases}$

Therefore:

$$L[i,j] = \begin{cases} L[i-1, j] \\ L[i, j-1] \end{cases}$$ > which ever is larger!

$$\Rightarrow L[i,j] = \max(L[i-1,j], L[i,j-1]).$$