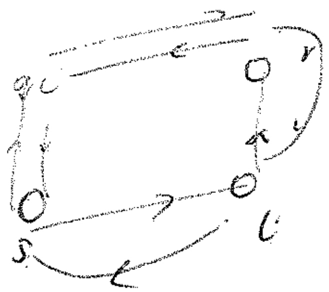
Elements of Dynamic Programming

- Optimal substructure
- Bottom up
- Subproblems shortest & longest paths



- Longest path does not have independent subproblems.
- Overlapping subproblems, Reconstructing an optimal soln

Longest common subsequence

- Biological & genomic applications
- Subsequence, $X = \langle x_1, x_2, \ldots x_m \rangle$
  $Z = \langle z_1, z_2, \ldots z_k \rangle$
- Common subsequence $\langle i_1, i_2, \ldots i_k \rangle$ $i_1 < i_2 \ldots i_k$
  $x_{i_j} = z_j$

- longest common subsequence.

- Brute force approach

- Optimal substructure

  Let $X_i = \langle x_1, x_2, \ldots x_i \rangle \quad X_0 = \epsilon$

  examples

  Thm: Let $X = \langle x_1, \ldots x_m \rangle \quad Y = \langle y_1, \ldots y_n \rangle$

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

  if $x_m = y_n$, then $z_k = x_m = y_n$ & $Z_{k-1} = LCS(X_{m-1}, Y_{n-1})$

  if $x_m \neq y_n$ and $z_k \neq x_m \rightarrow Z = LCS(X_{m-1}, Y_n)$

  if $x_m \neq y_n$ and $z_k \neq y_n \rightarrow Z = LCS(X, Y_{n-1})$

Let $c[i,j]$ = length of LCS of sequences X_i and Y_j

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

~~if c[i,j)~~

	y_j	B	D	C	A	B	A
x_i	0	0	0	0	0	0	0
A	0	0	0	0	↖1	←1	↖1
B	0	↖1	←1	←1	↑1	↖2	←2
C	0	↑1	↑1	↖2	←2	↑2	↑2
B	0	↖1	1↑	2↑	2↑	↖3	←3
D	0	1↑	↖2	2↑	↑2	↑3	↑3
A	0	↑1	↑2	↑2	↖3	3↑	4↖
	0	↖1	2↑	2↑	3↑	↖4	4↑

if $c[i-1,j] \geq c[i,j-1]$

$b[i,j] = \uparrow$

else

$b[i,j] = \leftarrow$