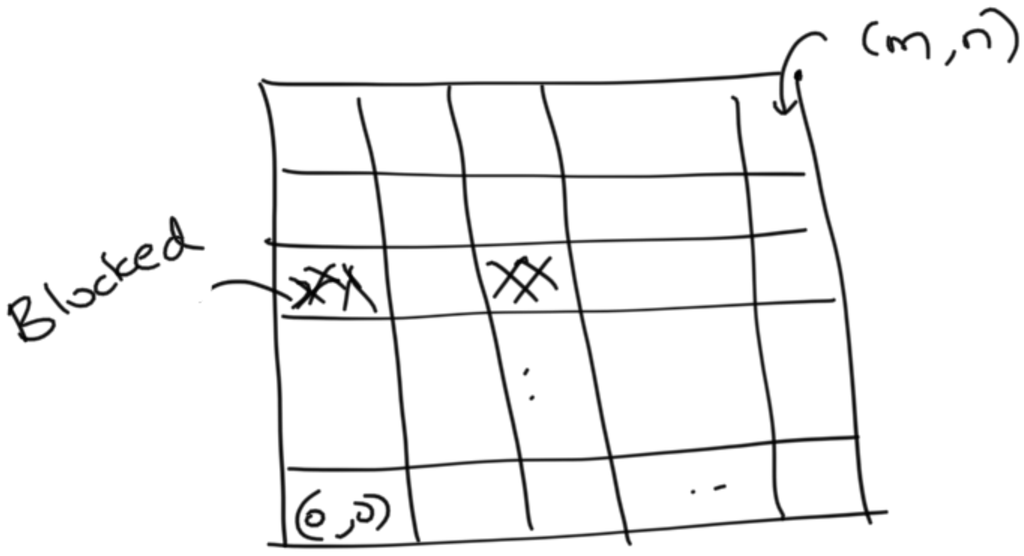
DP problems



How many paths from (0,0) to (m-1, n-1) if only allowed to go 'UP' or 'RIGHT'?

No blocks: need m Rights, n UPs,

so $\binom{m+n}{n} = \binom{m+n}{m}$

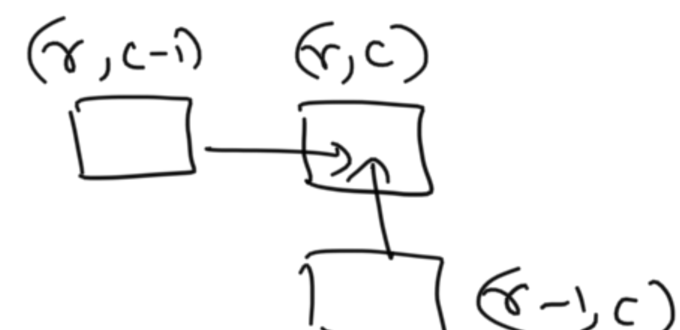
If blocks: let $H(i,j)$ = # of paths from (0,0) to (i,j).

$$H(r,0) = 1 \quad \forall\ 1 \le r \le m$$

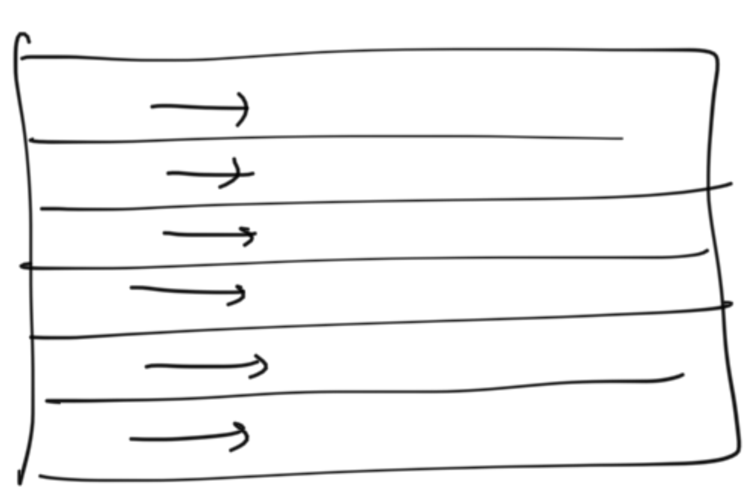
$$H(0,c) = 1 \quad \forall\ 1 \le c \le n$$

$$H(0,0) = 1$$

$$H(r,c) = \begin{cases} H(r,c-1) \\ \quad + H(r-1,c) \\ \text{if no block at } (r,c) \\ \\ 0 \quad \text{otherwise} \end{cases}$$



compute



Row wise

$H(i,j)$ row wise/ col wise..

Longest common substring.

straw
tabra

$u = u_0 u_1 \cdots u_{m-1}$
$v = v_0 v_1 \cdots v_{n-1}$

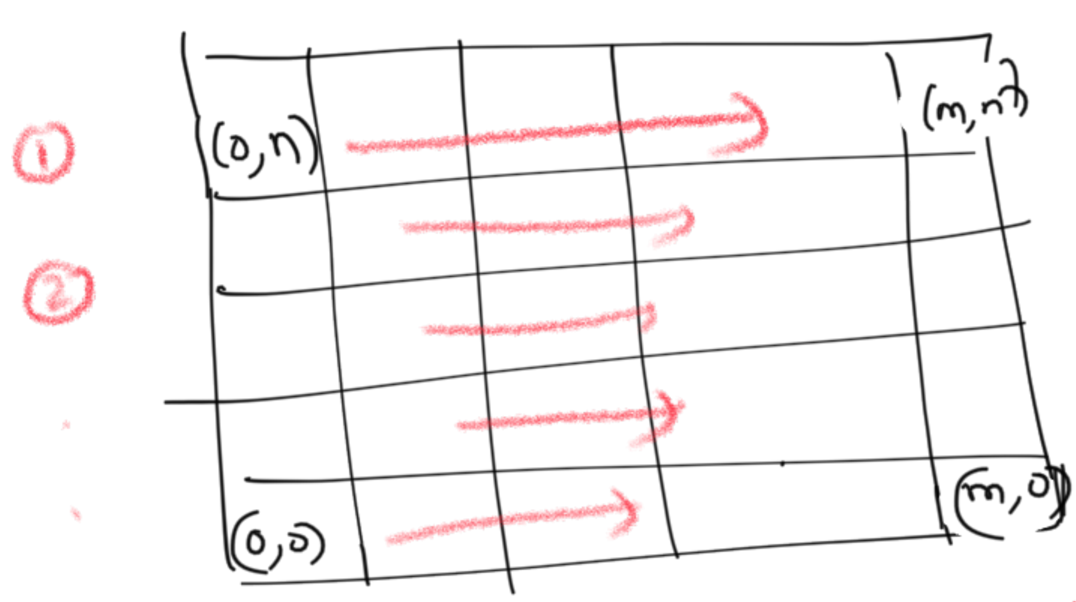
$L(i,j)$ = length of longest common substring

$\overrightarrow{u_i \cdots\cdots}$
$\overrightarrow{v_j \cdots\cdots}$

starting at u_i in u and v_j in v

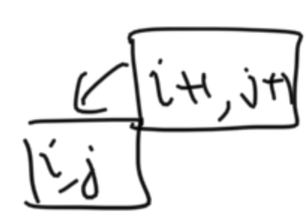
$L(m,j) = 0 \quad \forall\ j \le n$
$L(i,n) = 0 \quad \forall\ i \le m$

$$L(i,j) = \begin{cases} 1 + L(i+1, j+1) & \text{if } u_i = v_j \\ 0 & \text{if } u_i \neq v_j \end{cases}$$



Fill up row wise $L(i,j)$ ...

dependency



Read off common substring by

$\underbrace{u_i u_{i+1} \cdots}_{L(i,j) \text{ length}}$

Application: 'diff' in LINUX, genetic similarity bet species measuring

Longest common subsequence

$u = u_0 u_1 \dots u_{m-1}$

$v = v_0 v_1 \dots v_{n-1}$

straw

tabrat

(gaps allowed in bet) ..

$$L(u,v) = \begin{cases} 1 + L(u_1 \dots u_{m-1}, v_1 \dots v_{m-1}) & \text{if } u_0 = v_0, \\ \max\begin{pmatrix} L(u_0 \dots u_{m-1}, v_1 \dots v_{n-1}) \\ L(u_1 \dots u_{m-1}, v_0 v_1 \dots v_{n-1}) \end{pmatrix} & \text{if } u_0 \neq v_0 \end{cases}$$

$$L(i,j) := L(u_i \dots u_{m-1}, v_j \dots v_{n-1})$$

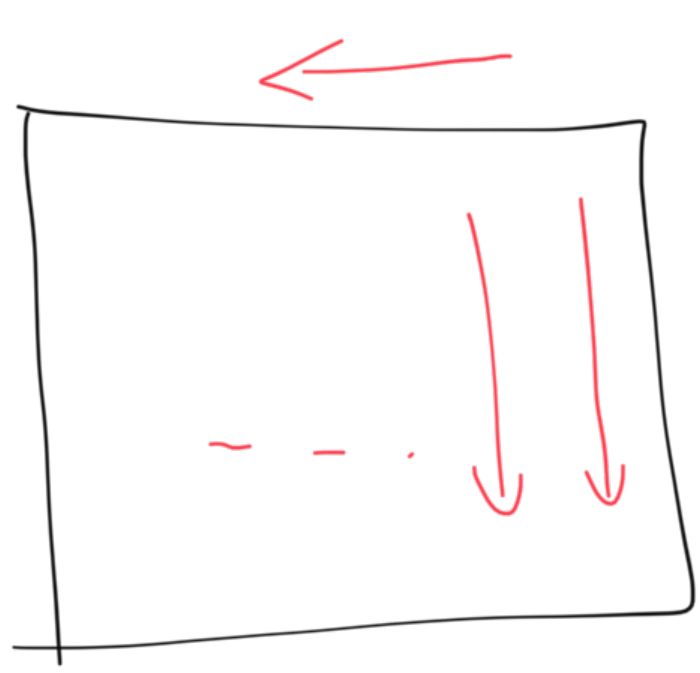
$$L(i,j) = \begin{cases} 1 + L(i+1, j+1) & \text{if } u_i = v_j \\ \max\left(L(i,j+1), L(i+1,j)\right) & \text{if } u_i \neq v_j \end{cases}$$

$L(m, j) = 0 \ \forall j$

$L(i, n) = 0 \ \forall i$

(i,j)

$i, j+1$	$i+1, j+1$
(i,j)	$(i, j+1)$



for instance, can Fill up $L(i,j)$ col by col..

Ans = $L(0,0)$

To find longest common subseq, see if $L(0,0)$ was got by $u_0 = v_0$ or $u_0 \neq v_0$. (and if so which max)...

Edit distance (Levenshtein)

string u

string v

$$u \xrightarrow{\text{transform}} v$$

edit operations $\begin{cases} \text{insert characters} \\ \text{delete characters} \\ \text{substitute characters} \end{cases}$

edit distance(u, v) = min (# of edit operations needed to make u into v)

Applications: spell corrector (find word in dictionary @ min edit distance)

compare similarity of genes of species...

Longest common subseq
LCS

b i s e c t

s e c r e t

$$\text{b i s e c t} \xrightarrow[\text{bi}]{\text{delete}} \text{sect} \xrightarrow[\text{re}]{\text{insert}} \text{sec re t}$$

edit distance without substitution $= l(u) - LCS(u, v)$

$+ \; l(v) - LCS(u, v)$

let $ED(i, j)$ = edit distance bet
$(u_i \dots u_{m-1}, \; v_j \dots v_{n-1})$

$u_i \;\; u_{i+1} \dots \;\; u_{m-1}$

$v_j \;\; v_{j+1} \dots \;\; v_{n-1}$

if $u_i = v_j$, $\;\; ED(i, j) = ED(i+1, j+1)$

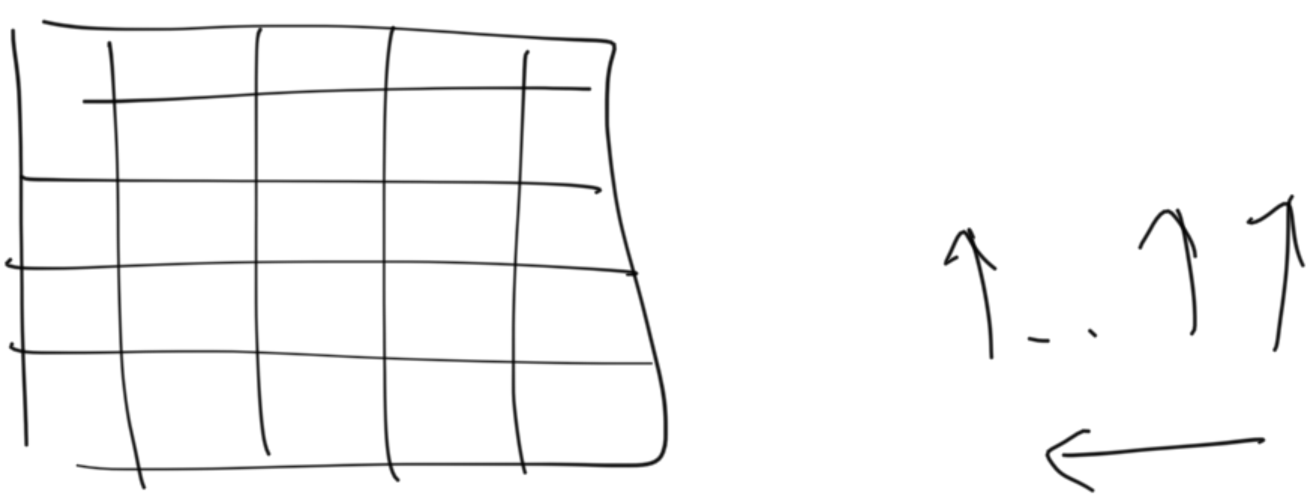
if $u_i \neq v_j$ $\quad ED(i,j) = \min \begin{pmatrix} 1 + ED(i+1, j+1) & \text{A} \\ 1 + ED(i+1, j) & \text{B} \\ 1 + ED(i, j+1) & \text{C} \end{pmatrix}$

A	B	C
$\overset{v_j}{\cancel{u_i}}$ $u_{i+1} \dots$ u_{m-1}	$\cancel{u_i}$ $u_{i+1} \dots$ u_{m-1}	$v_j\, u_i$ $u_{i+1} \dots$ u_{m-1}
v_j $v_{j+1} \dots$ v_{n-1}	v_j $v_{j+1} \dots$ v_{n-1}	v_j $v_{j+1} \dots$ v_{n-1}
substitute u_i with v_j	delete u_i	insert v_j before u_i

$ED(m+1, j) = n - j$ $\quad u = \phi$, $v = v_j \dots v_{n-1}$ — insert all letters

$ED(i, n+1) = m - i$ $\quad u_i \dots u_{m-1}$, ϕ — delete all letters

Ans: $ED(0,0)$



Trace back path to recover how exactly $u \to v$??

* $O(mn)$ complexity, $O(mn)$ storage
* can be efficient about space usage.

if filling col by col, to fill current col, only need prev col.

So only need to store 2 cols at one time

[can trace back sol. also by keeping track of how sol. is building up...]