Given 2 strings $a = a_1 a_2 \cdots a_m$
$b = b_1 b_2 \cdots b_n$

find longest string that is subseq of both a & b.

e.g. 0 1 0 1 3 1 2 1 — 10312
1 0 3 2 0 1 0 2

Define subproblems: for $i = 0, \ldots, m$, $j = 0, \ldots, n$

let $C(i,j)$ = length of the LCS of $a_1 \cdots a_i$ & $b_1 \cdots b_j$.

Want $C(m,n)$.

Recursive formula:

Case 1. sol'n not use a_i: $\Rightarrow C(i,j) = C(i-1,j)$

Case 2. sol'n not use b_j $\Rightarrow C(i,j) = C(i,j-1)$

Case 3. sol'n uses a_i & b_j with $a_i = b_j$

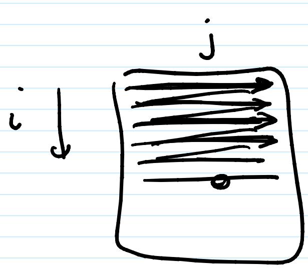
$\Rightarrow C(i,j) = C(i-1,j-1) + 1$

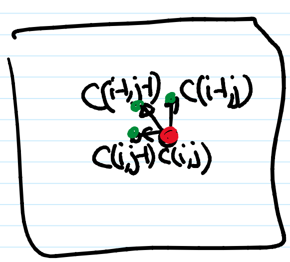
don't know which case, so try all & take max

$$\Rightarrow C(i,j) = \begin{cases} \max\{\cancel{C(i-1,j)}, \cancel{C(i,j-1)}, C(i-1,j-1)+1\} & \text{if } a_i = b_j \\ \max\{C(i-1,j), C(i,j-1)\} & \text{if } a_i \neq b_j \end{cases}$$

$(C(i-1,j) \le C(i-1,j-1)+1)$

Evaluation order: increas. order of i
for same i, increas. order of j





Run time: # table entries/subproblems $O(mn)$
time per entry $O(1)$

$\Rightarrow$ total time $\boxed{O(mn)}$

Space: $\boxed{O(mn)}$

Can reduce to $\boxed{O(n)}$ by remembering only last 2 rows if you just want the opt value

How to output opt sol'n:

output-sol(i,j):

--- (base case) ---

if $C[i,j] = C[i-1,j-1]+1$ & $a_i = b_j$

  output-sol$(i-1,j-1)$, append $a_i=b_j$ to output

else if $C[i,j] = C[i-1,j]$

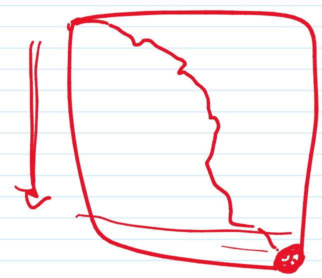
  output-sol$(i-1,j)$

else if $C[i,j] = C[i,j-1]$

  output-sol$(i,j-1)$

call output-sol(m,n)

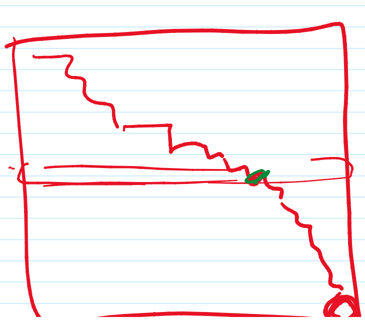
$O(m+n)$ additional time

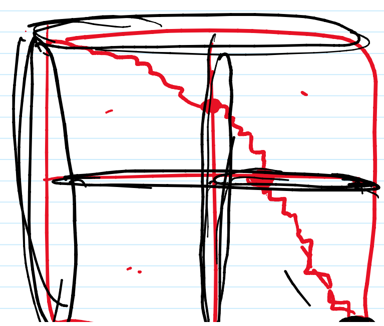


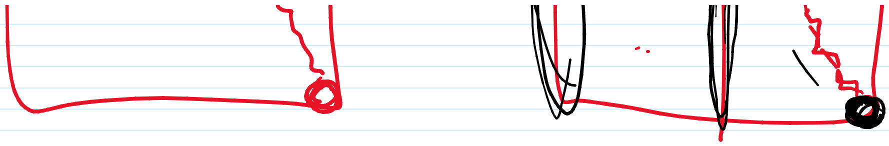
but this requires $O(mn)$ space

Space Improvement (Hirschberg '75 / Chowdhury, Ramachandran '06)

suppose $m=n$.







idea - divide & conquer

$$T(n) = 3\,T\left(\frac{n}{2}\right) + O(n^2)$$

$$\xrightarrow{\text{Master Thm}} \boxed{O(n^2)} \text{ time}$$

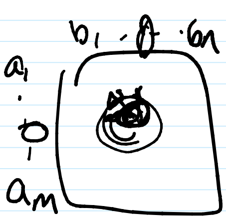
$$S(n) = S\left(\frac{n}{2}\right) + O(n)$$

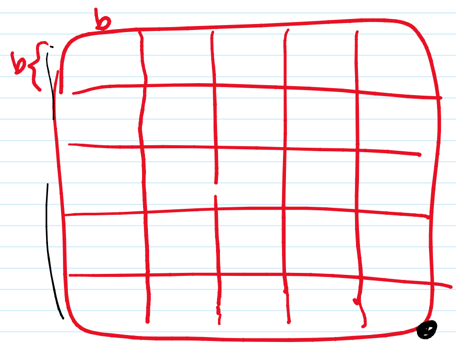
$$\Longrightarrow \boxed{O(n)} \text{ space}$$

Time Improvement (Masek & Paterson '80)

Assume $|\Sigma|$ const.

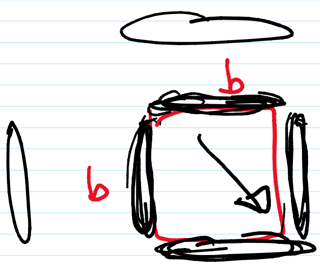
idea.





Four Russians trick

DP $\Rightarrow \left(\frac{n}{b}\right)^2$ calls to function



input: first row, first column
output: last row, last column

naively $O(b^2)$ time

total time $\Rightarrow O\left(\left(\frac{n}{b}\right)^2 \cdot b^2\right) = O(n^2)$

Precompute answers for all possible subproblems of size $b \times b$ & Store in table

how many diff. $b \times b$ subproblems?

$$|\Sigma|^b \cdot |\Sigma|^b \cdot 2^b \cdot 2^b$$
$$\leq |\Sigma|^{4b} \leq n^{\varepsilon}$$

Set $b = \frac{\varepsilon}{4} \log_{|\Sigma|} n \Rightarrow |\Sigma|^{4b} \leq |\Sigma|^{\varepsilon \log_{|\Sigma|} n} = n^{\varepsilon}$.

precomputation
$O(n^{\varepsilon} \cdot b^2) \ll O(n)$.

total time $O\left(\left(\frac{n}{b}\right)^2\right) = O\left(\frac{n^2}{(\log n)^2}\right)$ for const $|\Sigma|$

Rmk- extend to arb. $|\Sigma|$.

Open- substantially better than n^2?
e.g. $n^{1.99999}$?

Thm (Abboud et al.'15, Bringmann-Künnemann'73)

If there is an $O(n^{1.99999})$ time alg'm for LCS, then there is a "faster" alg'm for k-SAT
exponential-time

("Strong Exp. Time Hypothesis")
Stronger version of $P \neq NP$.