Announcements

Hw2 is available on Gradescope (one coding question and 2 written question. **Due Friday Feb 6** **only one late day.**

OK to import data structures

Problem 3: efficient = polynomial time

Prelim 1: Thursday, Feb 12. fill out this form, if you have a conflict.

Covers hw1-2, sections week 1-2, lectures through this week. Section next week is review. + DP quiz using HW2

Other prelim info and practice questions will be posted today (~ tonight)

Most TAs have their picture on the TA list on Canvas

Dynamic programming IV: Sequence Alignment

Correcting misspelled words:

- Acommodate

  Accomodate

- Seperate

  separate

- Reccommend
- Maintanence

  maitenance

The sequence alignment problem

sequence (of letters)

how similar are x & y

$x_1 \; \dots \; x_n$

$y_1 \; \dots \; y_m$

not allowed for today

δ = cost skipping a letter on one

$\mathrm{cost}(\alpha, \beta) > 0$ if $\alpha \neq \beta$ aligned

$\mathrm{cost}(\alpha, \alpha) = 0$

min total cost, no crossing

What may be good subproblem for Sequence Alignment?

What is last decision

$x_1 \dots x_n$

$y_1 \dots y_m$

Options for the end

- either match x_n y_m
- skip x_n
- skip y_m

as crossing not allowed

x_n

$\Rightarrow$ y_m

y_m skipped

Join by Web PollEv.com/evatardos772

What would be good subproblems for sequence alignment for aligning strings $x_1, \dots, x_n$ and $y_1, \dots, y_n$?

$x_1 \dots x_n$
$y_1 \dots y_m$
try to align

A. $O(i)$=min cost for aligning $x_1, \dots, x_i$ with $y_1, \dots, y_i$

B. $O(i)$=min cost for aligning $x_1, \dots, x_i$ with $y_1, \dots, y_n$

C. $O(i)$=min cost for aligning $x_1, \dots, x_n$ with $y_1, \dots, y_i$

D. Either B or C would work

E. None of these work

subproblem
$x_1 \dots x_{n-1}$
$y_1 \dots y_{m-1}$

Options
- either match x_n y_m
- skip x_n ← $x_1 \dots x_{n-1}$, $y_1 \dots y_m$
- skip y_m ← $x_1 \dots x_n$, $y_1 \dots y_{m-1}$

Subproblem for Sequence Alignment?

$Opt(i,j) =$ min cost of aligning $x_1 \dots x_i$, $y_1 \dots y_j$

$i = 0, \dots, n$
$j = 0, \dots, m$

$i,j \geq 1$

match x_i y_j

$$Opt(i,j) = \min\big(cost(x_i, y_j) + Opt(i-1, j-1),$$

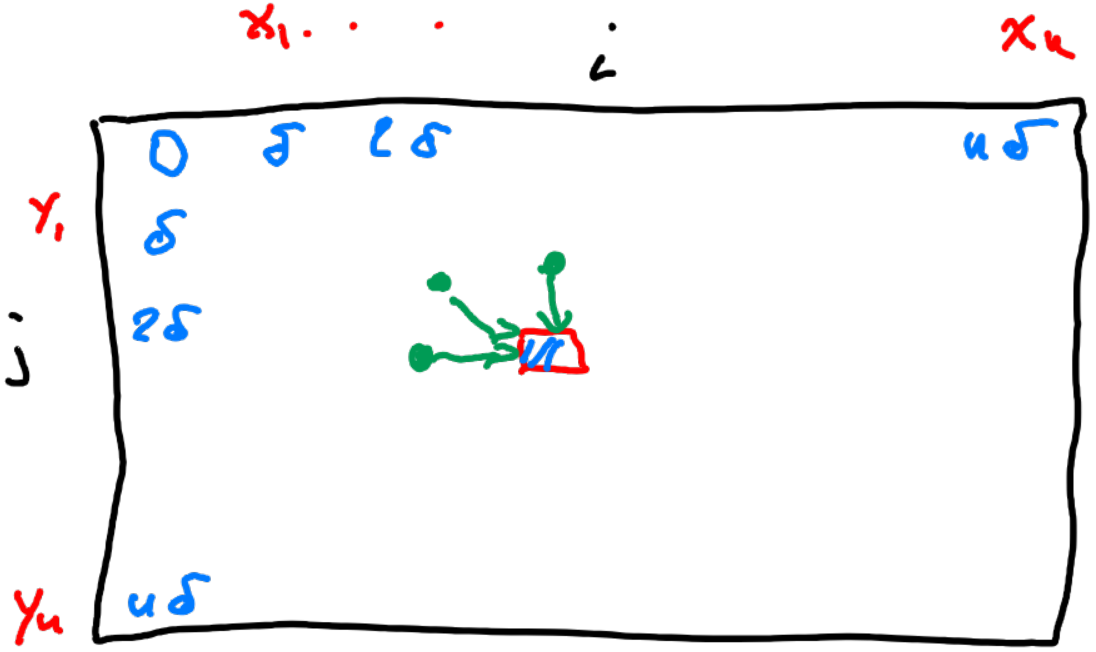
skip x_i: $\delta + Opt(i-1, j)$,

skip y_j: $\delta + Opt(i, j-1)$ $\big)$

base:

$Opt(0,j) = \delta j$

$Opt(i,0) = \delta i$



$Opt(i,j)$

cost skip $= \delta$

$cost(\alpha, \beta) \geq 0$ if $\alpha \neq \beta$

The dynamic program

For $i = 0, \dots n$

$\quad Opt(i, 0) = \delta i$

For $j = 0, \dots m$

$\quad Opt(0, j) = \delta j$

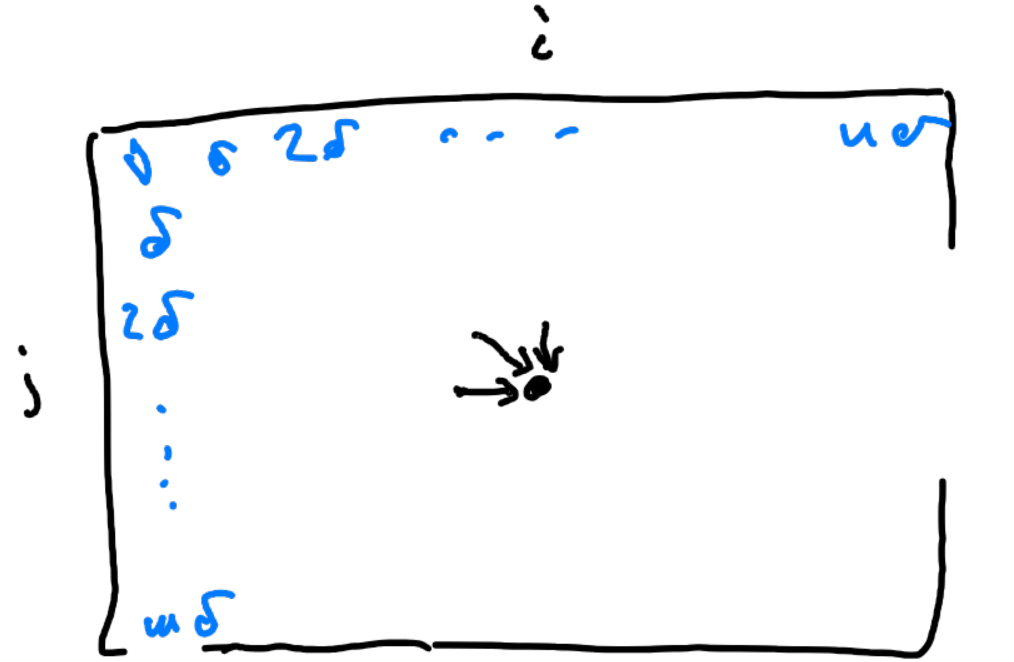
For $j = 1, \dots m$

$\quad$ For $i = 1, \dots n$

$$Opt(i,j) = \min\big(cost(x_i\, y_j) + Opt(i-1, j-1),\ \delta + Opt(i-1, j),\ \delta + Opt(i, j-1)\big)$$

end for

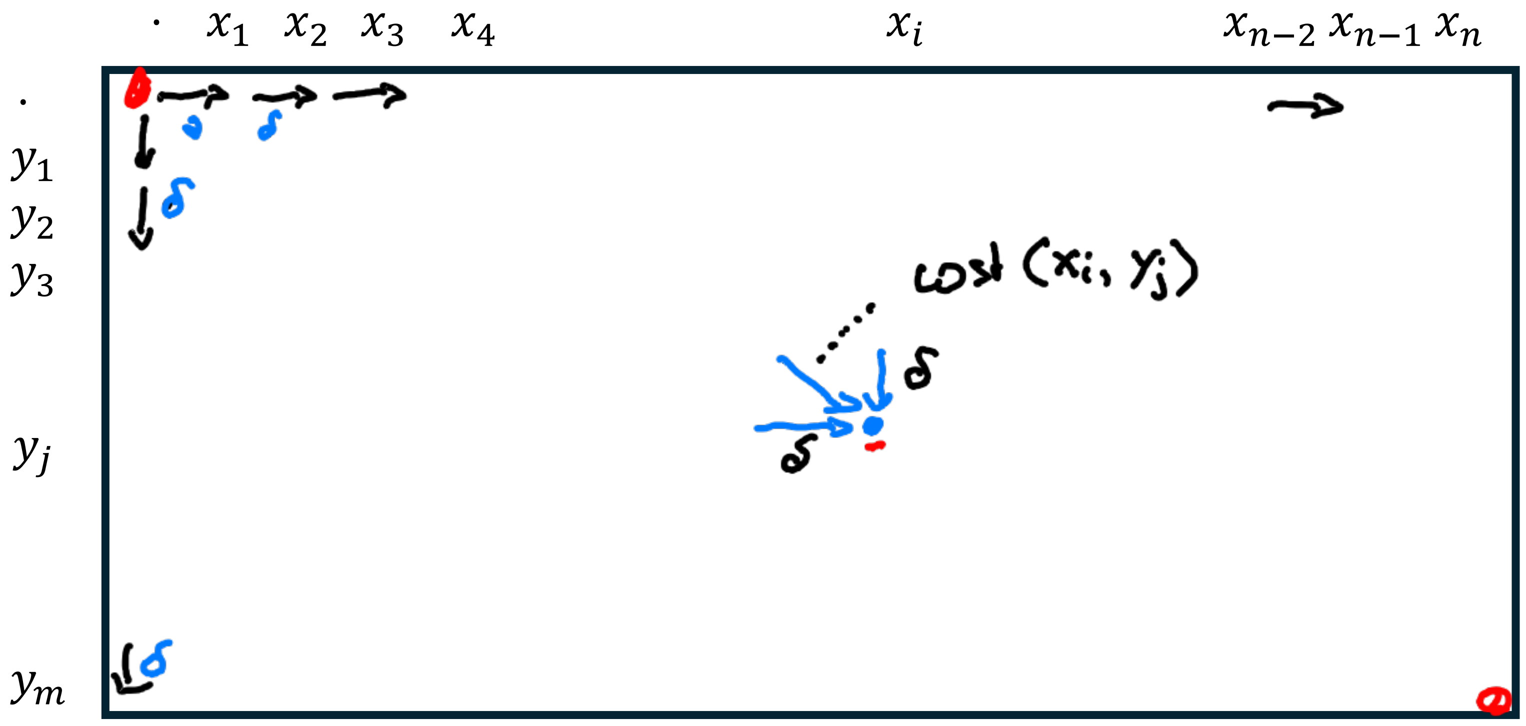
Return $Opt(n, m)$



Running time
$O(n \cdot m)$ size table

total time $O(mn)$

Sequence Alignment as min-cost path



G with $(n+1)(m+1)$ nodes

edges	cost
$(i-1,j) \to (i,j)$	δ
$(i,j-1) \to (i,j)$	δ
$(i-1,j-1) \to (i,j)$	$cost(x_i, y_j)$

best solution = min cost path from $s=(0,0)$ to $(n,m)=t$

Correctness, running time, and extracting the alignment

If direct dynamic program
correctness induction proof

If using reduction:
correctness
Part 1: solution of real problem (alignment)
corresponds to a path

Part 2: any path corresponds to an alignment