Longest common subsequence

$s = s_1 s_2 \ldots s_m$

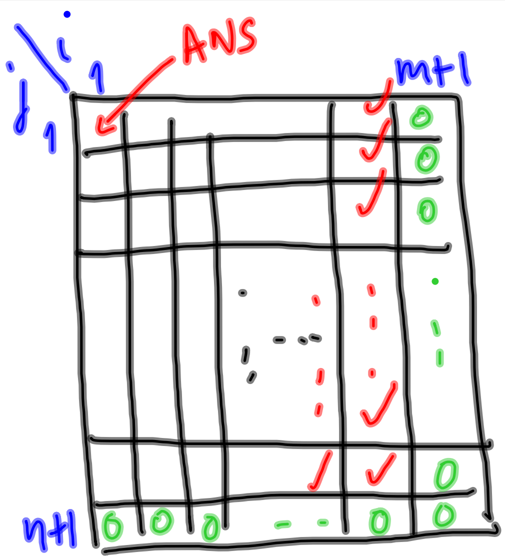
$t = t_1 t_2 \ldots t_n$

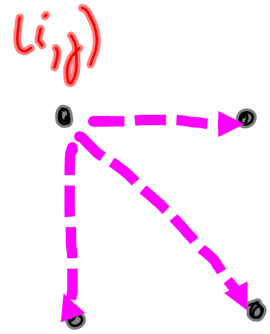
length of lcs

$llcs(i,j)$ length of lcs $s_i s_{i+1} \ldots s_m, t_j t_{j+1} \ldots t_n$

$llcs(m+1, j) = 0$, $llcs(i, n+1) = 0$

$$llcs(i,j) = \begin{cases} \text{if } s_i == t_j & 1 + llcs(i+1, j+1) \\ \text{else} & \max(llcs(i, j+1), \\ & \quad llcs(i+1, j)) \end{cases}$$





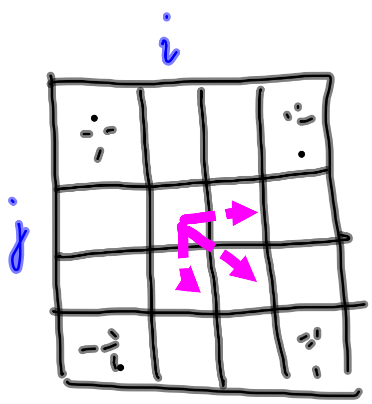
$$lcs(m,n) \begin{cases} 1 + 0 & \text{if } s_m == t_n \\ 0 = \max(0,0) & \text{if } s_m \neq t_n \end{cases}$$

Time: $O(m \cdot n)$

Witness?

Want lcs, not just llcs

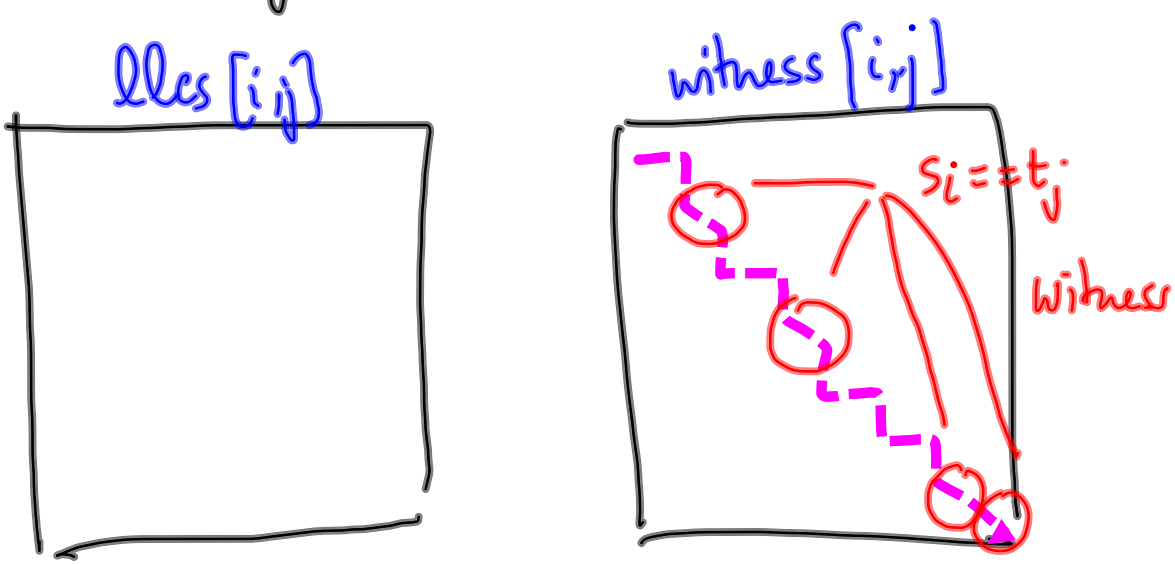
How was llcs(i,j) computed?



Record whether llcs(i,j) came from

llcs(i+1, j+1)
llcs(i+1, j)
llcs(i, j+1)

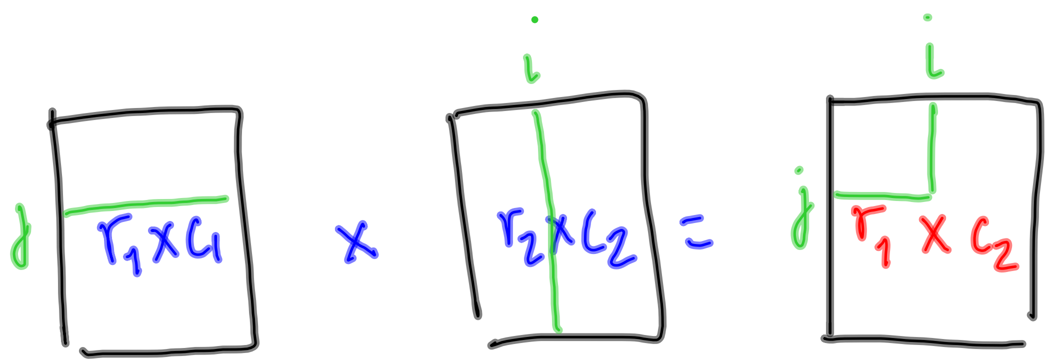
Maintain 2 grids



When computing llcs(i,j), set witness(i,j) = 1, 2 or 3 to record direction

Matrix multiplication

Complexity, assuming all arithmetic ops of unit cost



$c_1 == r_2$

How many entries? $r_1 \cdot c_2$

Each entry costs? $O(c_1) = O(r_2)$

$O(r_1 \cdot c_1 \cdot c_2) = O(r_1 \cdot r_2 \cdot c_2)$

Given $M_1 \times M_2 \times M_3$, all dimensions compatible,

should I compute $(M_1 \times M_2) \times M_3$

or $M_1 \times (M_2 \times M_3)$

or does it not matter?

100×1　$O(100^2)$

Takes $O(10^4)$　$O(100^2)$　100×100

$M_1 \times M_2 \times M_3$

100×1　1×100　100×1

1×1　$O(100)$　$O(10^2)$

$O(100)$　100×1

Given

$$M_1 \times M_2 \times \cdots \times M_n$$

$$r_1 c_1 \quad r_2 c_2 \quad \cdots \quad r_n c_n$$

find an optimal evaluation scheme

Each scheme corresponds to a "bracketing" of the expression

What is the final step?

$$(M_1 \times \ldots \times M_i) \times (M_{i+1} \times \ldots \times M_n)$$

r_1 c_i $c_i == r_{i+1}$ r_{i+1}, c_n

Assume split at i, last step cost is

$$r_1 \cdot c_i \cdot c_n = r_1 \cdot r_{i+1} \cdot c_n$$

To reach this point, we must have evaluated

$M_1 \times M_2 \times \ldots \times M_i$

$M_{i+1} \times \ldots \times M_n$

Cost of final split at i is

$$\text{Cost}(1..i) + \text{Cost}(i+1..n) + r_1 \cdot c_i \cdot c_n$$

Which i?

Minimize over all $i \in \{2, .., n-1\}$

Recursively compute cost for $M_1 \times .. \times M_i$

split at M_j

$M_1 \times M_2 .. M_j$ $\underbrace{M_{j+1} \times .. \times M_i}_{\text{general subproblem}}$

$Cost(i, j)$ Optimum (min) cost of multiplying $M_i \times M_{i+1} \cdots \times M_j$

In general, compute all splits at $k \in \{i, \ldots, j-1\}$ and evaluate $Cost(i, k)$, $Cost(k+1, j)$

Fix $k \Rightarrow$ final mult costs $r_i \cdot c_k \cdot c_j$

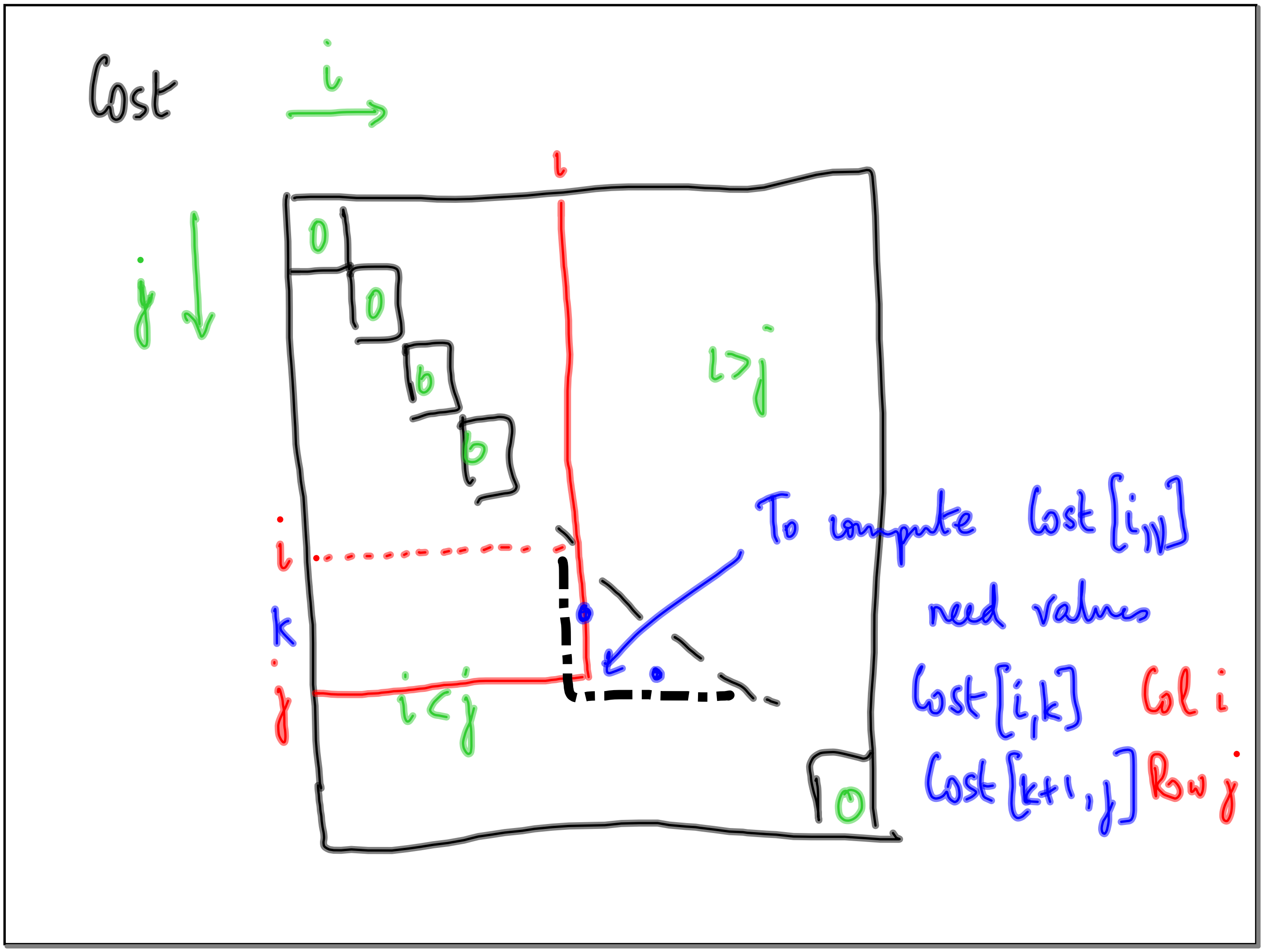
$$Cost(i,j) = \min_{k \in \{i,\ldots,j-1\}} \left[Cost(i,k) + Cost(k+1,j) + r_i \cdot c_k \cdot c_j \right]$$

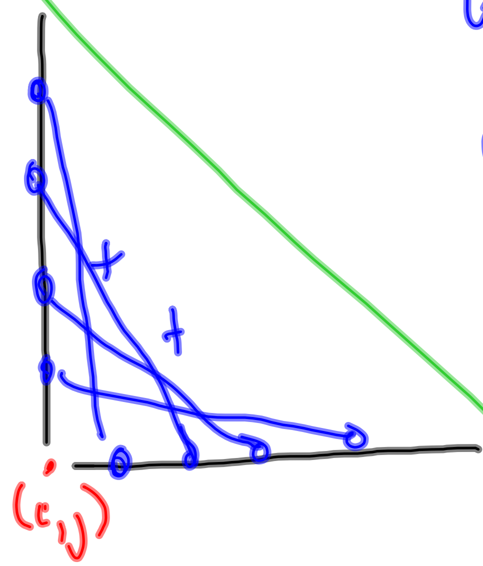
$$Cost(i,i) = 0$$

$$Cost(i,i+1) \Rightarrow k=i \Rightarrow \overset{0}{Cost(i,i)} + \overset{0}{Cost(i+1,i+1)} + \boxed{r_i c_i c_{i+1}}$$

$k \quad l$

$$M_1 \times M_2 \times \dots \times M_i \times \dots \times M_j \times \dots \times M_n$$


Cost
i
j
0
0
0
0
i
i > j
i
k
j
i < j
0
To compute Cost[i,j]
need values
Cost[i,k] Col i
Cost[k+1,j] Row j

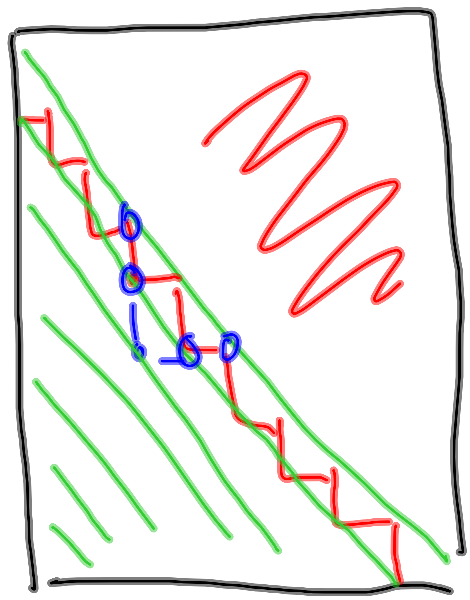


Effort of filling in cost[i,j] is proportional to $j-i$

$O(n^2)$ entries, $O(n)$ work per entry

$\Rightarrow O(n^3)$ overall

In what order should we fill up the matrix



Witness?

For each (i,j), record a choice k
that minimizes the cost

As in lcs, compute $Cost(i,j)$ & $Witness(i,j)$
in parallel

Recover witness from $Witness(i,j)$