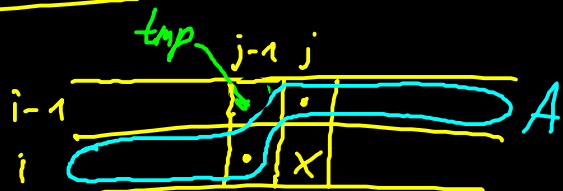


Exercise 3

Problem 7



procedure ScoresLine(S, T):

for $j = 1, \dots, n$

$A[j] = j \cdot g$

for $i = 1, \dots, m$

$tmp = (i-1) \cdot g$

for $j = 1, \dots, n$

$new = \begin{cases} tmp + p(S_i, T_j) \\ A[j] + g \\ g + \begin{cases} i \cdot g, & j = 1 \\ A[j-1], & j > 1 \end{cases} \end{cases}$

$tmp = A[j]$

$A[j] = new$

return A

procedure Score(S, T)

$A := \text{ScoresLine}(S, T)$

return $A[n]$

Space: array $A \rightsquigarrow O(n)$ entries

$O(n)$ temporary variables

Runtime: - initialisation of $A: O(n)$

• "filling out" the matrix $M \rightsquigarrow O(m \cdot n)$

76)

Observation 1: To align S, T , $m = |S|$, $n = |T|$

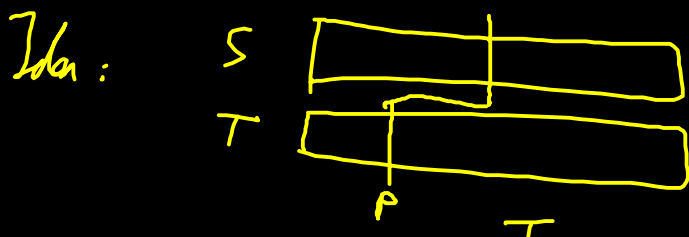
\rightsquigarrow a matrix of size $m \times n$ is enough
(eliminate first row and column)

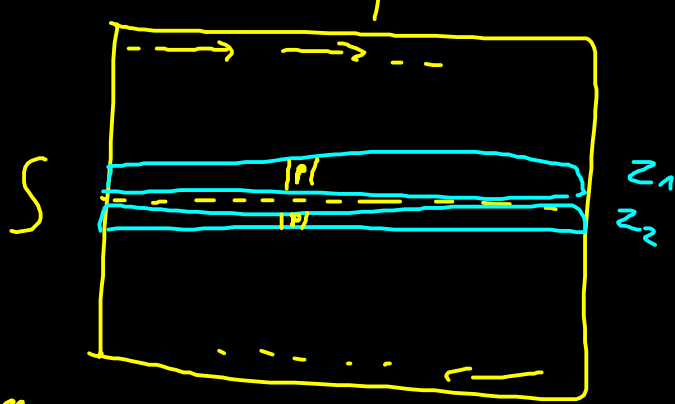
Observation 2:



We can fill

out the alignment matrix from bottom to top.





- Fill out upper half from top to bottom and
 - lower half from bottom to top
- $z_1[p]$ will contain the score of an optimal alignment of $S_{1, \frac{m}{2}}$ and $T_{1, p}$
- $z_2[p]$ will contain the score of an optimal alignment of $S_{\frac{m}{2}+1, m}$ and $T_{p+1, m}$

→ minimize $z_1[p] + z_2[p]$ by one iteration over $1, \dots, m$

- Fill recursively optimal alignments of $S_{1, \frac{m}{2}}$ and $T_{1, p}$ and $S_{\frac{m}{2}+1, m}$ and $T_{p+1, m}$

- Stop recursion if $|S| = 1$
 - $|T| = 0$ → deletion

- $|T| > 0 \rightarrow$ iterate over $i = 1, \dots, |T|$,
calculate $\delta(S_n, T_i)$, fill minimum m
- If $m > 2g \rightarrow$ 1 deletion, $|T|$ insertions
- If $m \leq 2g \rightarrow$ replace that one character,
insert the $|T|-1$ others

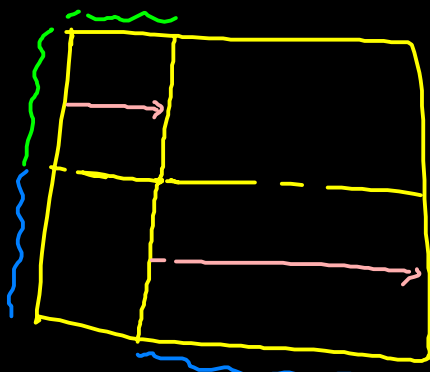
• output alignment immediately

\rightarrow works if we do the "left" recursion first

Running time

- First level of recursion
 $\rightarrow m \times n$ Matrix, fill out in $m \cdot n$ constant time steps
calculates p in $n+1$ steps

- Second level of recursion
 $\rightarrow \frac{m}{2} \cdot p + \frac{m}{2} \cdot (n-p)$
 $= \frac{m \cdot n}{2}$ cells to fill out



calculates p in $(p+1) + (n-p+1) = n+2$ step

⋮

$ll(m)$ recursion levels

$$\Rightarrow T = \sum_{i=0}^{ll(m)} \left[\frac{m \cdot n}{2^i} + n + 2^i \right] \leq m \cdot n + n[ll(m)+1] + 2m - 1$$

$$= O(m \cdot n)$$

Line complexity: • Any (see a): $O(m)$

• recursion length $h(m) \rightsquigarrow O(\log(m))$

$\Rightarrow O(m + \log(m))$

Problem 8

DSSP \leq_p MASP

- NP: a) guessing a solution $\rightsquigarrow O(l \cdot k)$ where $L = \sum_{i=1}^k |S_i|$
- b) compute the score $\rightsquigarrow O(l \cdot k^2)$

• $S_1, \dots, S_k \in \{0,1\}^*$, $m \in \mathbb{N}$

$$p(0,0) = p(1,1) = p(-,-) = 1$$

$$p(0,1) = p(1,0) = m \binom{k}{2} + 1$$

$$g = 1$$

$$d = m \binom{k}{2}$$

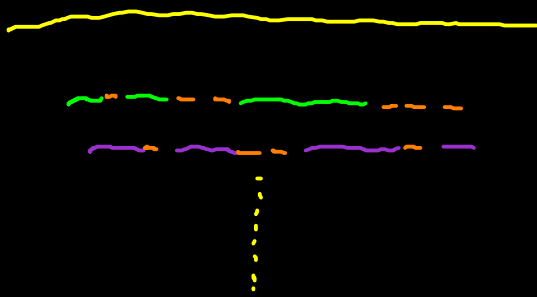
I_{DSSP} yes-instance $\iff f(I_{\text{DSSP}})$ yes-instance
for DSSP for MASP

\Leftarrow : MA with score $\leq d$ exists

\rightsquigarrow every column contributes $\binom{k}{2}$ to the total score
Since there can't be any (real) substitution

\rightsquigarrow construct unique consensus from the MA
 (every column contains only $\{-, 0\}$ or $\{-, 1\}$)
 which is also a supersequence for S_1, \dots, S_k

\Rightarrow we have supersequence for S_1, \dots, S_k of length $\leq n$

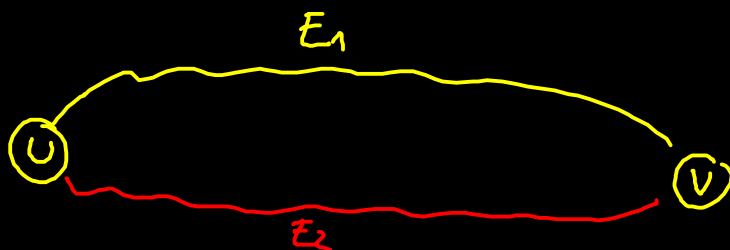


This sequence allows immediately the construction of a
 MA of S_1, \dots, S_k without mismatches.

$$g) \quad \left(g^*(u, v) = g(u, v) - (\xi(u) - \xi(v)) \right)$$

Let $E_1 \subseteq E$, $E_1 = \{(u_1, v_1), \dots, (u_k, v_k)\}$

be arbitrary shortest path in G w.r.t g^*



Suppose that path $E_2 \neq E_1$ is shorter w.r.t g

$$E_2 = \{ (v'_1, v'_2), \dots, (v'_{n-1}, v'_n) \}$$

$$\Rightarrow \sum_{(v_i, v_{i+1}) \in E_1} g(v_i, v_{i+1}) > \sum_{(v'_i, v'_{i+1}) \in E_2} g(v'_i, v'_{i+1})$$

$$\Rightarrow \sum \left(g^*(v_i, v_{i+1}) + (\xi(v_i) - \xi(v_{i+1})) \right) > \sum \left(g^*(v'_i, v'_{i+1}) + (\xi(v'_i) - \xi(v'_{i+1})) \right)$$

telescopic

\Rightarrow

sum

$$\sum \left(g^*(v_i, v_{i+1}) + \xi(v_i) - \xi(v_{i+1}) \right) > \sum \left(g^*(v'_i, v'_{i+1}) + \xi(v'_i) - \xi(v'_{i+1}) \right)$$

$$\Rightarrow \sum_{E_1} g^*(v_i, v_{i+1}) > \sum_{E_2} g^*(v'_i, v'_{i+1})$$

E_1 is shortest path wrt g^*