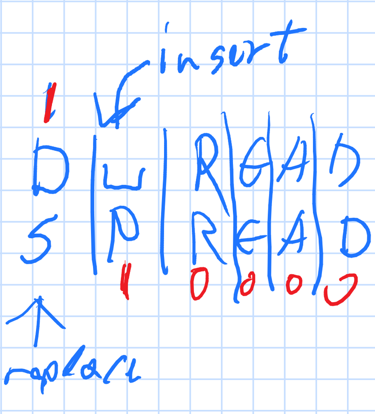
374 Lecture 15 10/11/22

- Edit distance
- CYK

DREAD
SPREAD

DREAD
SPREAD] 2 edit operations

- Delete a letter
- Insert a letter
- replace a letter



HAR-PELED
SHAPEYED

4

PEA|CE
PRI|CE

αx $\quad \alpha, \beta \in \Sigma^*$ $\quad x, y \in \Sigma$
βy

optimal alignment

α | x αx | α | x
β | y β | y βy |

$x=y$ ignore   insert   deletion
$x \neq y$ change

$S[1 \ldots m]$
$T[1 \ldots n]$ input strings

$$f(i,j) = \begin{cases} j & i=0 \\ i & j=0 \\ \min \begin{pmatrix} f(i-1, j-1) + [S[i] \neq T[j]] \\ f(i, j-1) + 1 \\ f(i-1, j) + 1 \end{pmatrix} \end{cases}$$

zero if equal
one otherwise

$f(i,j)$ is the min edit distance between $S[1 \ldots i]$ and $T[1 \ldots j]$

$f(i,j)$ $\quad S[1 \ldots m]$ $\quad T[1 \ldots n]$
$\quad i = 0, 1, \ldots m$ $\quad j = 0, 1, \ldots n$

$(n+1)(m+1) = O(nm)$

---

explicit memorization

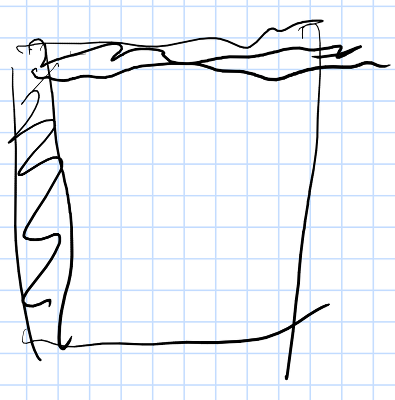
```
M[0... m][0...n] <- +∞
int ed(i,j):
    if i=0 then return j
    if j=0 then return i
    if M[i][j] is finite then return M[i][j]
    x <- f(i-1, j-1) + [S[i] ≠ S[j]]
    y <- f(i-1, j) + 1
    z <- f(i, j-1) + 1
    M[i][j] <- min(x, y, z)
    return M[i][j]
print ed(m,n)
```

Analysis

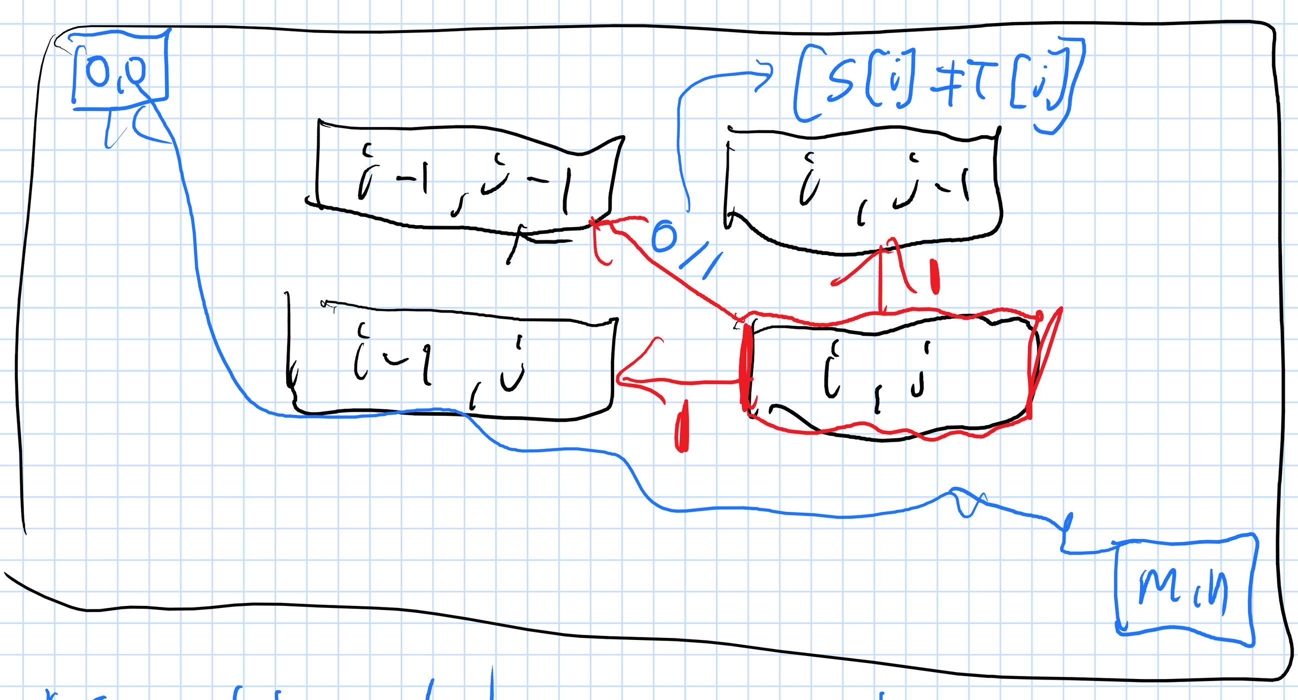
$O(mn) * O(1) = O(mn)$ Running time/space

Dynamic program for edit distance

```
M[0... m][0...n] : table
For i=0 to m do   M[i][0] <- i
For j=0 to n do   M[0][j] <- j
for i=1 to m do
    for j=1 to n do
```

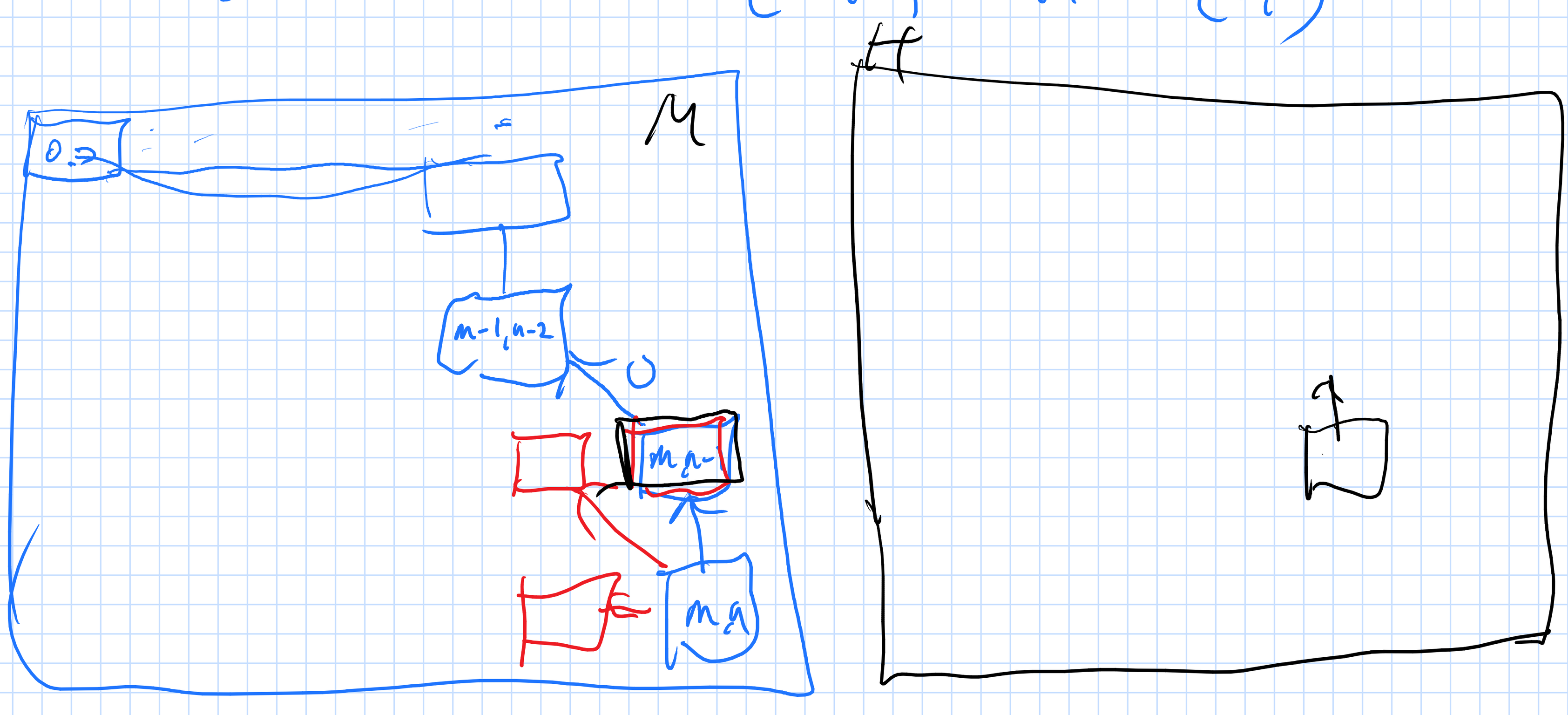


$$M[i,j] \leftarrow \min\Big(M[i-1, j-1] + [S[i] \neq S[j]],\ M[i-1, j] + 1,\ M[i, j-1] + 1\Big)$$



DAG directed acyclic graph.

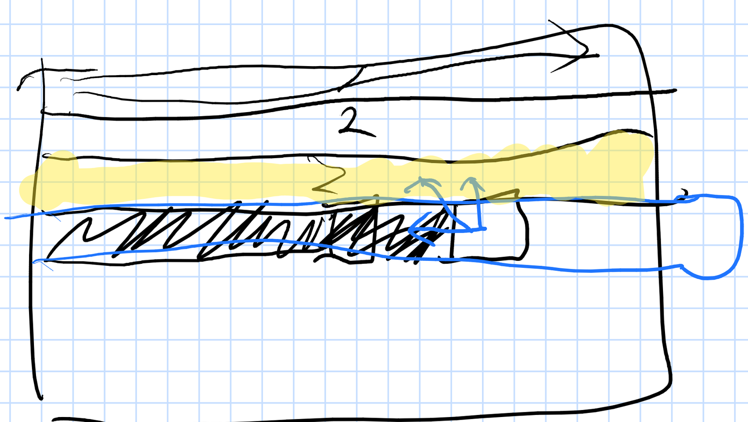
Problem becomes: Compute shortest path in DAG between (m,n) and (0,0)



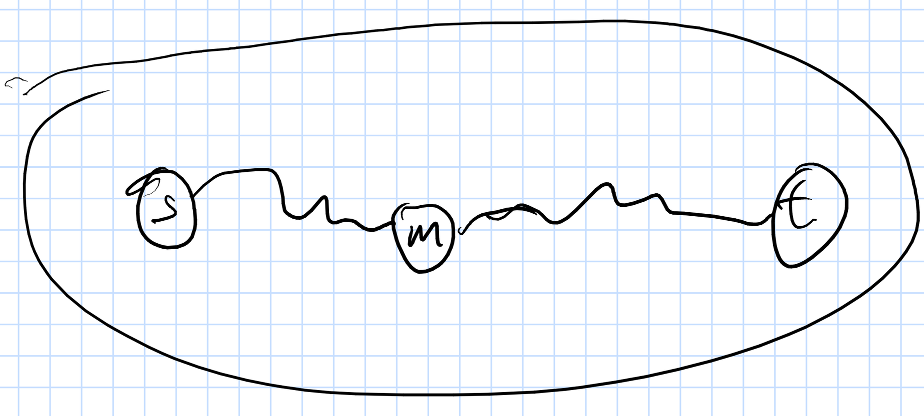
shortest path in DAG can be computed in linear time in the size of the graph.

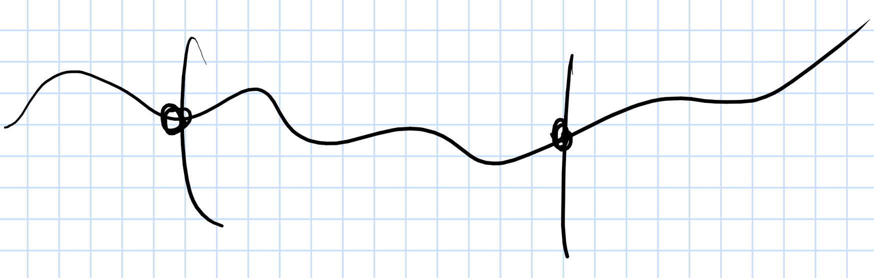
Longest path can be computed for DAGs in linear time in the size of the graph

---

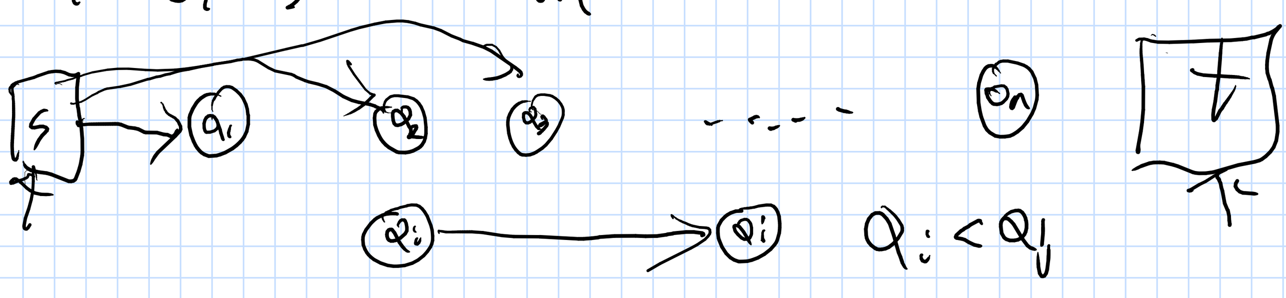


O(m) space





$a_1, a_2, a_3 \ldots a_n$



$a_i < a_j$

$O(n^2)$