# Dynamic Programming

- used in DNA sequence alignment

A C G T C T A G

A C – T C T A G

↑

not aligned

- how to align sequences?
- Come up with number indicating

- sequence alignment:
  based (partially) on
  Levenshtein distance
  used in Unix diff
  program.

  e.g. diff prog.cpp old.cpp

  really useful for versioning
  (SVN depends on it)

- toy example

WHAT
WH-Y
| | |
I + | - δ - μ

WHAT
WH-Y

cost

Use scores for

identity (match) +1

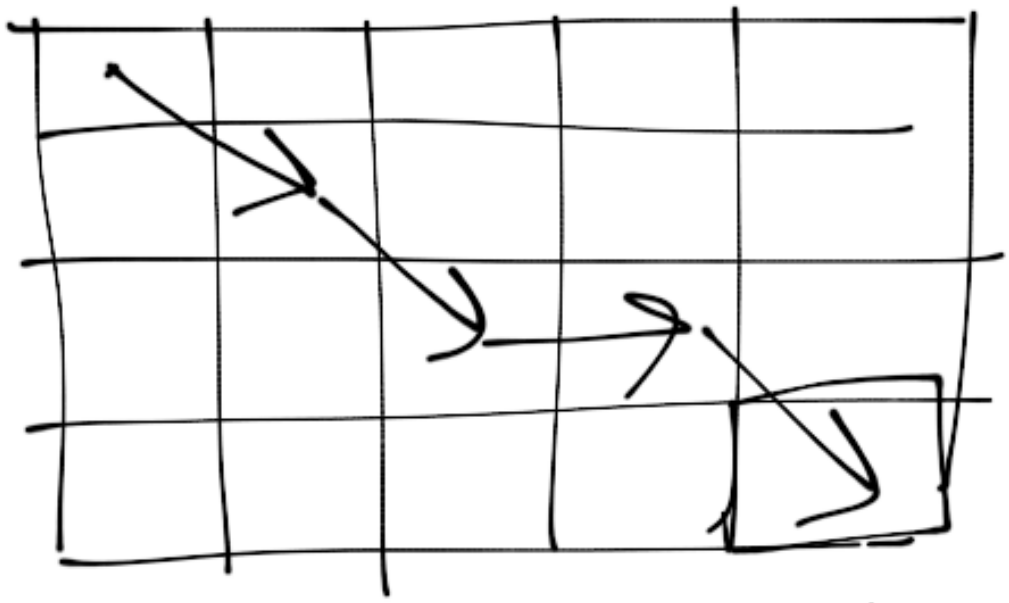
substitution (mismatch) - μ

insertion / deletion (indel) - δ

- Use high enough low scores for mismatches

- rewrite adjacency as a matrix: search space ($S_1$)

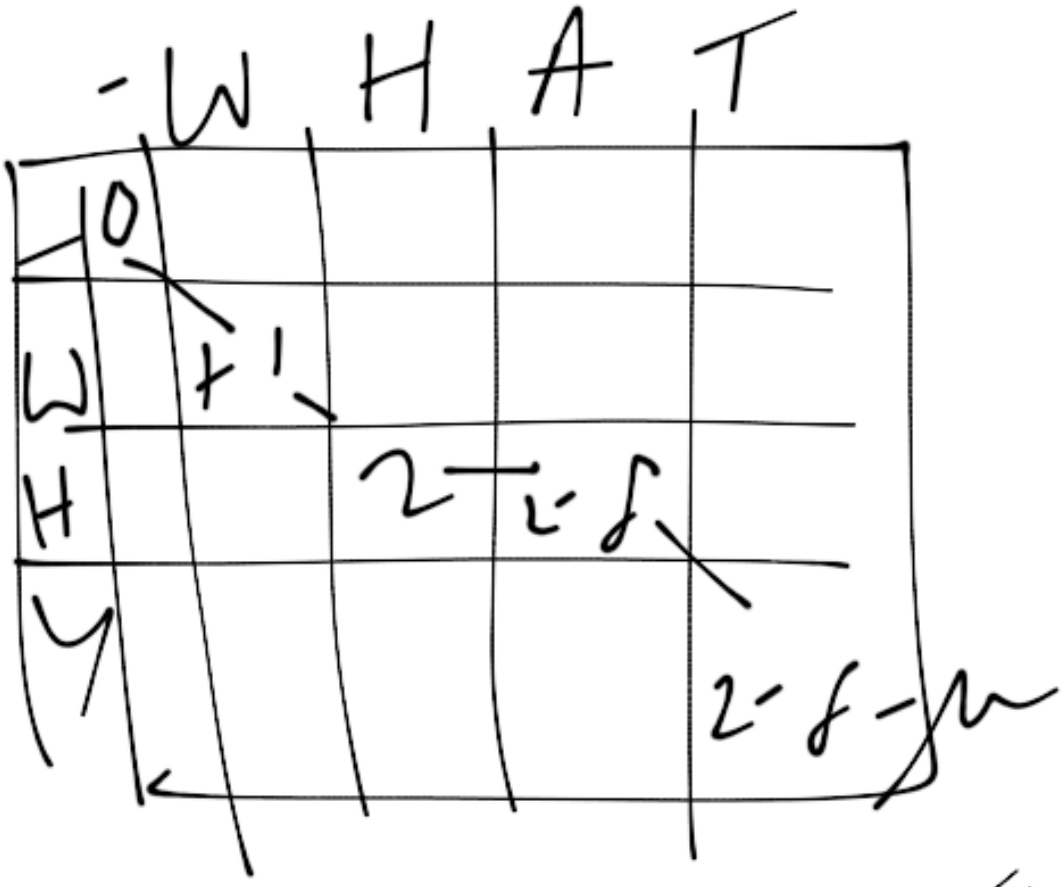
- WHAT



target W
eg. H
($S_2$) Y

dest.

- adj. starts at upper left, proceeds right, down, or diag. depending on state (stored in matrix)



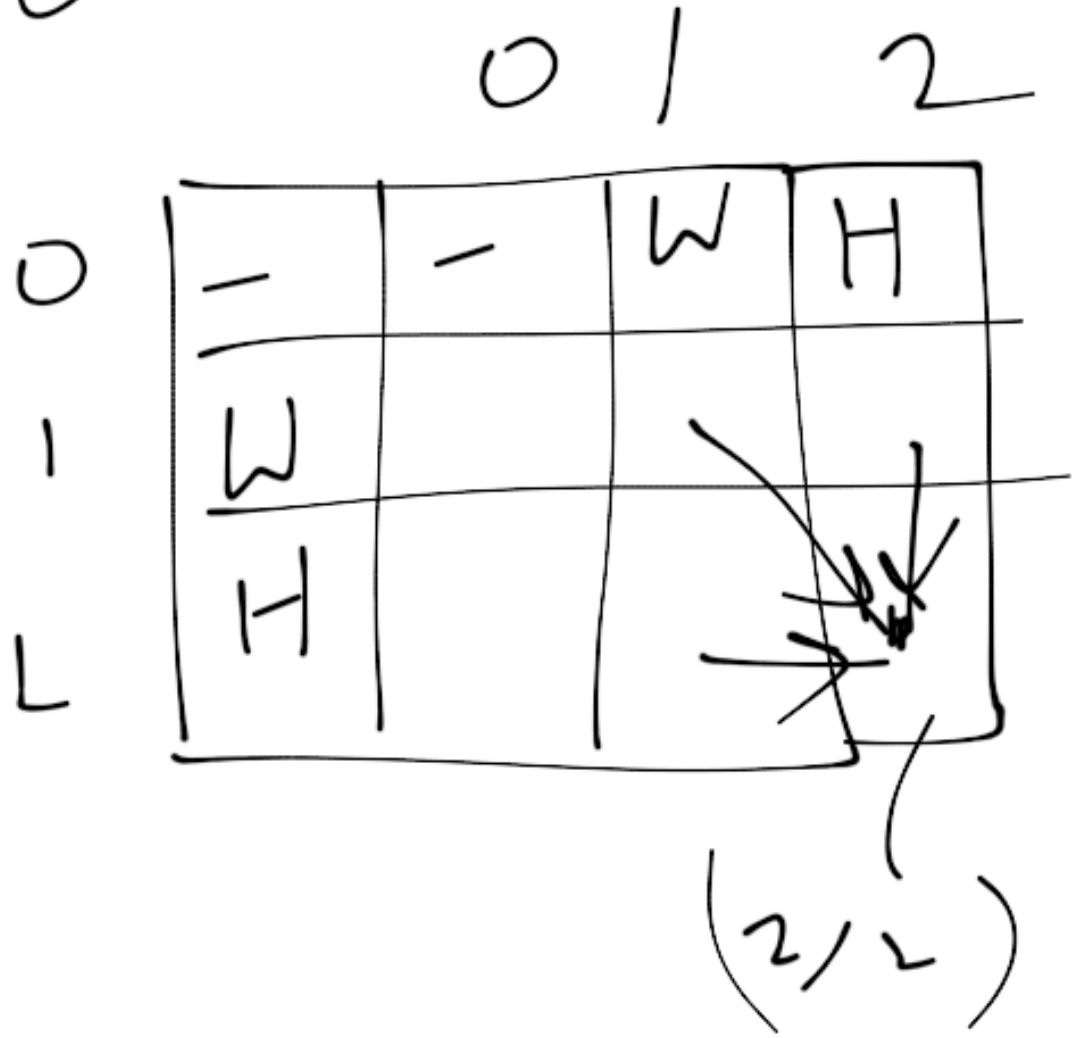

- once matrix is filled in, work backward, from destination to get alignment

- say we have three costs:

identity : +1

sub : −1

indel : −2



- to get to (2,2):

a) from (2,1) (better to H)

WHAT
WHY
W-HY

$S_{2,2} = S_{2,1} - 2$

(instead)

b) from (1,2) (choose)

W-HAT
WHAT
WHY

$S_{2,2} = S_{1,2} - 2$

PICK C

(instead)

c) - from (1,1)

WHAT
WHY

$S_{2,2} = S_{1,1} + 1$

- alg.:

Init first row, first col with $g + i$ (-2)

Linear gap cost

matrix C...

| | - | W | H | A | T |
|---|---|---|---|---|---|
| - | 0 | -2 | -4 | -6 | -8 |
| W | -2 | | | | |
| H | -4 | | | | |
| A | -6 | | | | |

- calculate scores for all cells starting at top-left:

```
for (i = 1; i <= rows; i++)
    for (j = 1; j <= cols; j++)
        c0 = C[i-1][j-1] +
             s(s1[i-1], s2[j-1])
        c1 = C[i-1][j] + g
        c2 = C[i][j-1] + g
        C[i][j] = max(c0, c1, c2)
```

s(s1[i-1], s2[j-1]):

s(a, b):

if (a == b)

return

(match cost)

else

return -1

(substitution cost)

Needleman-Wunsch alg.