CT

12/3/19

FAT NOTES by Karina

speedy boi RDB (who is dating a 2nd year so ouch my bones).

e.g.
Seq 1: ACG
Seq 2: AGT

+2	match
-1	mismatch
-2	gap

Recursion:

$al(3,3) = \underline{al(2,2)} + al(3,2) + al(2,3)$
$al(2,2) = al(1,1) + al(2,1) + al(1,2)$
$al(3,2) = al(2,1) + al(3,1) + \underline{al(2,2)}$
$al(2,3) = al(1,2) + \underline{al(2,2)} + al(1,3)$

recursion bois

look at this grim repetition
↳ wasted computation
↳ this is why you'd want to use dynamic programming

kinda gross ngl

Needleman-Wunsch

Recursion cont...

$f(<>, <ACG, AGT>)$ → + another option (match T against gap)

$f(<G,T>, <AC, AG>)$ $f(<G,->, <AC, AGT>)$

3 recursive options 3 recursive options

overlapping subproblems = bad

NEEDLEMAN-WUNSCH

RELEVANT TO COURSEWORK!!!

or just use arr, except Python is a string so maybe not

Dynamic Programming:
- make a cool matrix my dude

Backtrack matrix → finds alignment

Scoring matrix:

finds best score

	–	A	C	G
–	0	← -2	← -4	← -6
A	↑ -2	↖ 2	← 0	← -2
G	↑ -4	↑ 0	↖ 1	↖ 2
T	↑ -6	↑ -2	↑ -1	↖ 0

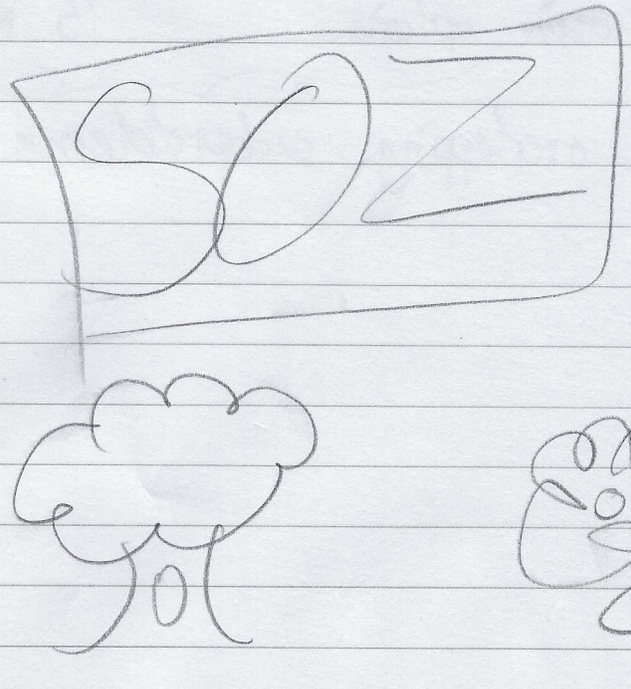
best score here jam

	–	A	C	G
–		↖		
A		Diag ←	←	↖
G		↑	↑	↑
T		↑	↑	↑

- basically looking for max score to put in each cell
  - is it better to match, mismatch, or match against a gap?

Alignment 1:
A C G
A G T
+2 -1 -1 = 0

Alignment 2:
A C G –
A – G T
+2 -2 +2 -2 = 0



Ends-free scoring:

- need to change algorithm so gaps at beginning + end of sequences are free

accidentally wrote man, whoops i is smart dab

$$s(m,j) = \max\{c(m,j) + s(m-1, j-1), s(m, j-1), s(m-1, j) - 2\}$$

$$s(i,n) = \max\{c(i,n) + s(i-1, n-1), s(i, n-1) - 2, s(i-1, n)\}$$