ALIGNMENT

The sequence alignment problem:

- given:
  - 2 sequences (X and Y)
  - scoring matrix (S)
- compute: the pairwise alignment of X and Y of MAXIMUM score

Global aligment ~ optimal alignment along the entirety of both sequences

For example: given: X = ACAAT, Y = TCAGAT

with scoring scheme:
- +0 for gap
- +0 for mismatch
- +1 for match

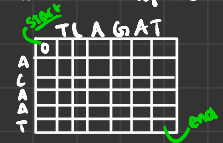
we could get the alignment:

T	C	A	G	A	T
A	C	A	-	A	T

score: 0+1+1+0+1+1 = 4

Thus, there are 3 possible alignments for a letter in a sequence:
- MATCH: align letter w/ same letter in other sequence $\binom{A}{A}$
- MISMATCH: align letter not w/ same letter $\binom{A}{T}$
- GAP/INDEL: align letter w/ gap $\binom{A}{-}$
  - * biological application of indels: an insertion/deletion mutation @ some point in evolutionary history

- There is a bijection (1:1 correspondence) between alignments of X and Y and directed paths from the top left cell (beginning) to bottom right cell (end) of edit graph



- the edit graph is a directed graph with edge weights
- max alignment score = max directed path from beginning → end

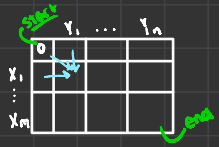
suppose sequence X is of size m and Y is of size n:

→ # of alignments btwn X and Y is exponential

However, this algorithm will find the optimal alignment in quadratic ($O(mn)$) time! hooray! what a beautiful algorithm! we love dynamic programming ♥

Now, for the algorithm: $X = x_1 x_2 x_3 \dots x_m$; $Y = y_1 y_2 y_3 \dots y_n$

- Edit graph:
  - dimensions: $(m+1)(n+1)$ (# of rows, # of columns; row = index of x, column = index of y)
  - entries have form (i, j)
    $1 \le i \le m$; $1 \le j \le n$



- edges: 3 types: horizontal, vertical, diagonal
  - vertical: gap in Y $(i-1, j) \to (i, j)$ $(x_i, -)$
  - horizontal: gap in X $(i, j-1) \to (i, j)$ $(-, y_j)$
  - diagonal: alignment (match | mismatch) $(i-1, j-1) \to (i, j)$ (x_i, y_j)
- $S(i,j)$ = score of the max score path from start to i, j

ex

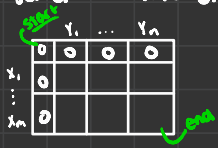
$i-1, j-1$	$i-1, j$
$i, j-1$	i, j

→ any optimal path from start → (i,j) must use one of the 3 green edges

Scoring scheme:

$$S(i,j) = \max \begin{cases} S(i-1, j) + \delta(x_i, -), & \text{cost of Y gap} \\ S(i, j-1) + \delta(-, y_j), & \text{cost of X gap} \\ S(i-1, j-1) + \delta(x_i, y_j) & \text{cost of aligning } x_i \text{ and } y_j \text{ (either match or mismatch)} \end{cases}$$

First, you must initialize the edit graph (depending on the scoring scheme - assume this one has +0 gap penalty)



Then, you can go cell by cell, calculating $S(i,j)$ based on the 3 surrounding cells.