1. Global alignment with a limited number of gaps [30 pts].
Suppose \( k \in \mathbb{Z} \) is a non-negative third input (in addition to input sequences \( v \) and \( w \)) to the alignment algorithm. Devise a dynamic programming algorithm that finds an optimal global alignment between \( v \) and \( w \) subject to the constraint that the alignment contains at most \( k \) blocks of consecutive indels. You may denote the scoring matrix by \( \delta \) which is pursuant to our notation in class.

2. Semi-global alignment [35 pts].
In this question, we consider a variant of the global alignment problem, where we impose no penalty on gaps at the end of one of the input sequences. Consider the following two alternative alignments:

\[
\begin{align*}
v: & \quad \text{CAGCA-CTTGATTCTCGG} \\
w: & \quad \text{---CAGCGTGG--------}
\end{align*}
\]
\[
\begin{align*}
v: & \quad \text{CAGCACTTGATTCTCGG} \\
w: & \quad \text{CAGC-----G-T----GG}
\end{align*}
\]

When doing global alignment under the scoring scheme of +1 for a match, -1 for a mismatch, and -1 for a gap, the second alignment is preferred despite our intuition that the first alignment is more biologically relevant. If the gaps on the ends of \( w \) are not penalized, then the first alignment scores higher. This approach is called semi-global alignment. Note that in this approach, the end gaps in one of the sequences \( (v) \) are penalized as in the standard global alignment. Show how to modify the Needleman-Wunsch algorithm to compute a semi-global alignment (including the initialization of the matrix, and the backtrace operations). Modify align.py to perform semi-global alignment and illustrate that your code is indeed performing semi-global alignment by running it on some examples and providing the output.

3. Local alignment [35 pts].
Modify the program align.py to perform local sequence alignment of proteins using a substitution matrix (here’s a link to a BLOSUM 62 matrix: http://www.ncbi.nlm.nih.gov/Class/FieldGuide/BLOSUM62.txt). Use your program to align the Aniridia protein from \( H. \ sapiens \) to the eyeless protein from \( D. \ melanogaster \) using the BLOSUM 62 substitution matrix. The
sequences of these proteins can be obtained from the Uniprot protein database using the following links: http://www.uniprot.org/uniprot/P26367 (aniridia) and http://www.uniprot.org/uniprot/018381 (eyeless). Note that Uniprot has an option for saving an entry in FASTA format.

Answer the following questions:

- What is the alignment and score you obtained?

- Align 100 randomly generated protein sequences with the same length and amino acid composition as the Aniridia protein with the eyeless protein. Compare the scores with that obtained above. Does that suggest a statistically significant similarity event?

* Optional: Quantify that statistical significance by providing a $p$-value. Any idea about the underlying null probability distribution?

Upload your answer on Canvas in one zip file or tarball. Include all the code/scripts you have written in your submission.