FIT3139 S2/2016: Assignment 2
(Due midnight 11:59pm on Sunday 16 October 2016)

[Weight: $10 = 1 + 2 + 1 + 5 + 1$ marks.]

This assignment tests your ability to reason about a real-world computational science problem. You will handle an optimization problem that is routinely handled by researchers around the globe. It will test your understanding of the dynamic programming paradigm, and your ability to design and implement it into a fully working program. Hope you will enjoy this exercise.

Supporting Material:

Use supporting material for your Week 9's Lab on Moodle to complete this assignment.

Follow these procedures while submitting this assignment:

The assignment should be submitted online via moodle strictly as follows:

- Fill out the Assignment Cover Sheet and submit as a part of your submission.
- All your written reports and scripts MUST contain your name and student ID.
- Use `gzip` or `Winzip` to bundle your work into an archive which uses your student ID as the file name. (STRICTLY AVOID UPLOADING `.rar` ARCHIVES!)
  - Your archive should extract to a directory which is your student ID.
  - This directory should contain a subdirectory for each of the five questions above, named as `q1/, q2/, q3/, q4/ and q5/`
  - Within each of those subdirectories, the contents must include reports and/or scripts corresponding to the respective questions.
  - All written reports should strictly be in PDF format with your name and student ID on each page.
- Submit your zipped file electronically via Moodle.

Background

In your lectures you were introduced a dynamic programming formulation for the problem of comparing and aligning any two sequences, given a substitution matrix S and a **subadditive (saturating) gap penalty function** of the form $\Gamma(l) = g_o + g_e \log(l)$, where l is the length of

a **run of gap symbols** anywhere in an alignment, g_o is the penalty incurred in opening a gap, and g_e is the penalty incurred in extending a (previously opened) gap. We discussed during the lectures that the time complexity of the resulting dynamic programming algorithm using $\Gamma(l)$ is $O(n^3)$, assuming two sequences being aligned have (roughly) n symbols each.

This assignment will consider the same problem of aligning two sequences, but replacing the above gap penalty function with **yet another subadditive gap penalty function** which is of the form: $\hat{\Gamma}(l) = g_o + g_e(l-1)$. *

Computing an **optimal** alignment of two sequences using the gap-penalty function $\hat{\Gamma}(l)$ allows for an efficient dynamic programming formulation with a time complexity of $O(n^2)$.

Using $\hat{\Gamma}(l)$ any alignment can be described as a string over **three** alignment states, one per each type of (possible) alignment columns:

1. The first state M (*Match*) represents any ungapped column in an alignment (i.e., a column without gap symbols).

2. The second state I (*Insert*) represents any column with a gap symbol ('$-$') in Sequence 1.

3. The third state D (*Delete*) represents any column with a gap symbol ('$-$') in Sequence 2.

For example, the alignment given below can be listed as a **string of states/columns**(see last row): M, D and I.

Column number	1	2	3	4	5	6	7	8	9	10	11	12	13
Sequence 1	A	–	–	S	Q	G	C	Y	V	E	–	–	–
Sequence 2	A	E	C	T	Q	G	–	–	W	E	S	T	H
State string	*M*	*I*	*I*	*M*	*M*	*M*	*D*	*D*	*M*	*M*	*I*	*I*	*I*

To achieve $O(n^2)$ time complexity with the gap penalty function $\hat{\Gamma}(l) = g_o + g_e(l-1)$, the main observation to note is that the subproblems are strictly additive **only between blocks** of alignment columns **of the same state**. (Do spend time exploring this observation.)

Using the example above, the given alignment can be partitioned into blocks of columns of same state as shown below:

Column number	1	2	3	4	5	6	7	8	9	10	11	12	13
Sequence 1	A	–	–	S	Q	G	C	Y	V	E	–	–	–
Sequence 2	A	E	C	T	Q	G	–	–	W	E	S	T	H
State string	*M*	*I*	*I*	*M*	*M*	*M*	*D*	*D*	*M*	*M*	*I*	*I*	*I*

If you look at the 3-state string (representing an alignment) in the last row from left-to-right, you will observe state transitions $M \to I$, $I \to I$, $I \to M$, $M \to M$ and so on. If you think about it, there are only 9 distinct state transitions possible. Each transition between

*It is suggested that while attempting this assignment you will benefit from doing the following:

1. Plot both $\Gamma(l)$ and $\hat{\Gamma}(l)$ for varying values of l and compare the growth of these two functions as l varies.

2. We have seen that $\Gamma(l)$ is subadditive during the lecture. Convince yourself that $\hat{\Gamma}(l)$ is also subadditive based on the information below.

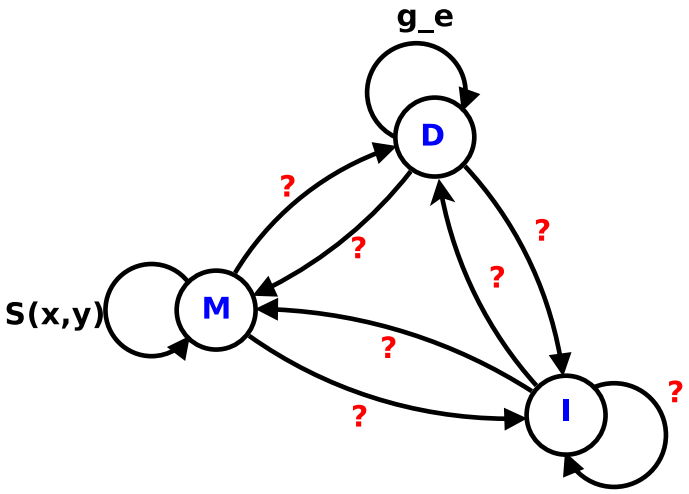

Figure 1: Finite state diagram of 9 possible transitions between states M, D and I when aligning any two sequences using a substitution matrix S and a gap penalty of the form $\Gamma(l) = g_o + g_e(l-1)$. Note: the scores associated with some transitions appear as edge labels in the figure. Others (shown with '?' symbols) have to be reasoned/answered as a part of this assignment (see Question 1 below)

alignment columns/states is one of these nine transitions. This can be best viewed as a finite state transition diagram shown in Figure 1, where nodes are alignment states and directed edges show transitions from one alignment state to another. Associated with each transition (edge) are labels representing either a score or a penalty for that transition. The values that these labels will take depend on the scoring/penalty function used for this problem.

In Figure 1 you will notice that a couple of edges are labeled with the score/penalty of the implied transition. At the top of the figure, you will see a labeled transition/edge going from $D \rightarrow D$. If you refer back to the example alignment in the previous page, you will notice that column 8 of the alignment corresponds to a transition of this kind. Appropriately, **given the gap penalty function** $\hat{\Gamma}(l)$, we are *extending* a previously opened gap in sequence 2 by one more gap symbol, and this would incur a penalty of g_e for that column of alignment. Similarly, the labeled transition shown in the left of the figure refers to the transition $M \rightarrow M$, which corresponds to the states of columns 5, 6 or 10 in the example alignment on the previous page. This appropriately shows a substitution score of $S(x, y)$, where x corresponds to the symbol in the first sequence, and y corresponds to the symbol in the second sequence. (For example, in column 5 of the alignment shown in the previous page, these letters involve Q in sequence 1 matched with Q in sequence 2.) In the same vein, **other state transitions will have their corresponding labels.**

Things fall into place here: we note that under this new gap penalty function, additivity holds within column blocks containing columns of the same state. Therefore, corresponding to Figure 1, a dynamic program can be defined as a recurrence relationship between three states over all the 9 possible transitions. This is realized by using three memoization (or history) matrices, one for **each** of the three states (block types). The recurrence rules of filling these matrices directly follow the finite state diagram shown in Figure 1.

More formally, let:

- $X = x_1 x_2 \cdots x_m$ and $Y = y_1 y_2 \cdots y_n$ be two given sequences.
- $S(x, y)$ be a substitution matrix that gives the score of matching any symbol x in sequence 1 with any symbol y in sequence 2.
- $\Gamma(l) = g_o + g_e(l-1)$ be the gap penalty function.

Using the labeled transitions in Figure 1, a **partially filled** dynamic programming recurrence relationships to compare two sequences is given by:

$$M_{i,j} = \max \begin{cases} M_{i-1,j-1} + S(x_i, y_j) \\ ? \\ ? \end{cases}$$

$$D_{i,j} = \max \begin{cases} D_{i,j-1} + g_e \\ ? \\ ? \end{cases}$$

$$I_{i,j} = \max \begin{cases} ? \\ ? \\ ? \end{cases}$$

Questions to be answered in this assignment

1. Complete and report the labels on the transitions in Figure 1, currently missing (marked by '?' symbols). (You can simply list these in your report in text format by listing the transition type followed by its label. For example: M→M: $S(x, y)$, D→D: g_e, etc.)
2. Complete and report the missing recurrence relationships associated with M, D and I history matrices in the dynamic programing above.
3. In the above dynamic program, the boundary conditions (i.e, scores in the first row/column of the history matrices M, D and I) are intentionally omitted. Report the boundary conditions for all these three matrices. Justifying your answer in your report.
4. Using the supporting material from week 9 lab, write a program that accepts as **arguments**: two sequences, a substitution matrix, and parameters g_o and g_e. The program should produces as output: (a) an optimal alignment, and (b) the corresponding optimal alignment score.
5. Run your program on **any five distinct pairs** of sequences given in the supporting material. Your report **MUST** contain the following information for each chosen pair:
   (a) the sequences being aligned.
   (b) the chosen values of g_o and g_e.
   (c) the optimal alignment score.
   (d) the optimal alignment of the sequences.

Do not hesitate to contact me if you need any clarification about this assignment.

```
-=o0o=-
  END
-=o0o=-
```