CS466: Introduction to Bioinformatics **Name: Nivedita Rethnakar**

Problem Set 1

Handed out: September 13, 2024 *Due: September 21, 2024 (11:59pm, CT)*

Instructions: This homework assignment consists of four questions worth a total of 50 points, question 5 is a bonus question for 6 points. These questions are based on the material covered in Lectures 1 to 5. **Do not forget to write your name at the top!**

1. **Asymptotic Running Time** [10 points]

   Consider the following running time functions, where $n > 0$.

3^n	n^3	$\sqrt{n}$	$n^2 \log(n)$	$n \log(n)$	$n!$	2^n	$n^2 \log(n!)$
$n(n+1) - n^2$	$n + n^2$	$n \log(n^2)$	$n^3 - n^2$	1	$n^2 - n$	n^n	$10{,}000$
$2^{n \log_2(n)}$	$n \log_3(3n)$	$2^{n \log_2(3)}$	$n^2 / \log(n)$	$n^{1.5}/n$	$n\sqrt{n}$	$n \log_{10}(n)$	$n^2/(100n)$

   a. **Identify groups** of functions such that for any pair $(f(n), g(n))$ of functions in the same group it holds that both $f(n) = O(g(n))$ and $g(n) = O(f(n))$. Note that some groups may contain a single function. [7 points]

   Hint: For example, $f(n) = 3n$ and $g(n) = n$ would be in the same group, as $f(n) = 3n = O(n) = O(g(n))$ and $g(n) = n = O(3n) = O(f(n))$.

   Group 1: 1, $10{,}000$
   Group 2: $\sqrt{n}$, $\dfrac{n^{1.5}}{n}$
   Group 3: n, $n(n+1) - n^2$, $\dfrac{n^2}{100n}$
   Group 4: $n \log n$, $n \log(n^2)$, $n \log_3(3n)$, $n \log_{10} n$
   Group 5: $n\sqrt{n}$
   Group 6: $\dfrac{n^2}{\log n}$
   Group 7: n^2, $n^2 - n$, $n + n^2$
   Group 8: $n^2 \log n$, $n^2 \log(n^2)$
   Group 9: n^3, $n^3 - n^2$
   Group 10: $n^2 \log(n!)$
   Group 11: 2^n
   Group 12: 3^n, $2^{n \log_2 3}$
   Group 13: $n!$
   Group 14: n^n, $2^{n \log_2 n}$

b. Arrange the resulting Big Oh running time groups in order from **fastest to slowest**. [3 points]

Group 1:	1, 10,000
Group 2:	$\sqrt{n}$, $\frac{n^{1.5}}{n}$
Group 3:	n, $n(n+1)-n^2$, $\frac{n^2}{100n}$
Group 4:	$n \log n$, $n \log(n^2)$, $n \log_3(3n)$, $n \log_{10} n$
Group 5:	$n\sqrt{n}$
Group 6:	$\frac{n^2}{\log n}$
Group 7:	n^2, $n^2 - n$, $n + n^2$
Group 8:	$n^2 \log n$, $n^2 \log(n^2)$
Group 9:	n^3, $n^3 - n^2$
Group 10:	$n^2 \log(n!)$
Group 11:	2^n
Group 12:	3^n, $2^{n \log_2 3}$
Group 13:	$n!$
Group 14:	n^n, $2^{n \log_2 n}$

2. **Sequence Alignment** [20 points]

Consider two DNA sequences **v** = GCACGC and **w** = AGCAATGGCCAAGGC. In this exercise, we will align the two sequences using a score of +1 for a match, -1 for a mismatch, and -1 for a insertion/deletion (i.e. a gap penalty of 1). We will use three different alignment algorithms. In each case, follow the specific instructions to provide requested information about the dynamic programming table or optimal alignment.

a. Consider the following global alignment of **v** with **w**.

v	-	G	C	A	-	-	-	-	C	-	-	-	G	-	C
w	A	G	C	A	A	T	G	G	C	C	A	A	G	G	C

(i) Give the score for this global alignment and **(ii) fill out** the dynamic programming table with the corresponding backtrace (i.e. highlight the corresponding path through this table and fill in cells on path with alignment scores). It suffices to only fill out cells that are part of the alignment. [4 points]

	-	A	G	C	A	A	T	G	G	C	C	A	A	G	G	C
-	0	-1														
G			0													
C				1												
A					2	1	0	-1	-2							
C										-1	-2	-3	-4			
G														-3	-4	
C																-3

Alignment score: -3

b. Consider the following fitting alignment. That is, an alignment of **v** and a substring **w′** of **w** with maximum global alignment score.

v	G	C	-	A	C	G	-	C
w′	G	C	C	A	A	G	G	C

(i) Give the score for this fitting alignment and **(ii) fill out** the dynamic programming table with the corresponding backtrace (i.e. highlight the corresponding path through this table and fill in cells on path with alignment scores). It suffices to only fill out cells that are part of the alignment. [4 points]

	-	A	G	C	A	A	T	G	G	C	C	A	A	G	G	C
-								0								
G									1							
C										2	1					
A												2				
C													1			
G														2	1	
C																2

Alignment score: 4

c. Consider the following dynamic programming table produced when finding an optimal fitting alignment. That is, an alignment of **v** and a substring of **w** with maximum global alignment score.

	-	A	G	C	A	A	T	G	G	C	C	A	A	G	G	C
-		0														
G			1													
C				2												
A					3											
C						2	1	0								
G									1							
C										2						

Give the fitting alignment corresponding to the highlighted path. [4 points]

Corresponding alignment:

v	G	C	A	A	-	-	G	C						
w’	G	C	A	C	C	C	G	C						

d. Consider the following local alignment (grayed out entries are not part of the alignment). That is, an alignment of a substring $\mathbf{v}'$ of $\mathbf{v}$ and a substring $\mathbf{w}'$ of $\mathbf{w}$ with maximum global alignment score.

$\mathbf{v}'$	G	C	A
$\mathbf{w}'$	G	C	A

(i) Give the score for this local alignment and **(ii) fill out** the dynamic programming table with the corresponding backtrace (i.e. highlight the corresponding path through this table and fill in cells on path with alignment scores). It suffices to only fill out cells that are part of the alignment. [4 points]

	-	A	G	C	A	A	T	G	G	C	C	A	A	G	G	C
-		0														
G			1													
C				2												
A					3											
C																
G																
C																

Alignment score: 3

e. Consider Σ to be the Latin alphabet, comprising all 26 capital letters. Let $\mathbf{v}$ = BIOLGIA and $\mathbf{w}$ = AOILGAI, utilizing local sequence alignment consider the below proposed alignment (with matches scored by +1 and everything else -1). In addition, consider the below table obtained using the Smith-Waterman algorithm discussed in class. **(i) Discuss whether the proposed alignment is an optimal local alignment, and (ii) enumerate/list all optimal local alignments.** [4 points]

A =

$\mathbf{v}'$	L	G	-	I	A
$\mathbf{w}'$	L	G	A	I	-

v \ **w**	0	A	O	I	L	G	A	I
-	-	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0
I	0	0	0	1	0	0	0	1
O	0	0	1	0	0	0	0	0
L	0	0	0	0	1	0	0	0
G	0	0	0	0	0	2	1	0
I	0	0	0	1	0	1	1	2
A	0	0	0	0	0	0	2	1

Is A optimal? No

Number of optimal alignments: 3

Optimal alignments (you may or may not need all boxes):

$\mathbf{v}'$	L	G									
$\mathbf{w}'$	L	G									

$\mathbf{v}'$	L	G	-	I							
$\mathbf{w}'$	L	G	A	I							

$\mathbf{v}'$	L	G	I	A							
$\mathbf{w}'$	L	G	-	A							

$\mathbf{v}'$											
$\mathbf{w}'$											

$\mathbf{v}'$											
$\mathbf{w}'$											

$\mathbf{v}'$											
$\mathbf{w}'$											

$\mathbf{v}'$											
$\mathbf{w}'$											

$\mathbf{v}'$											
$\mathbf{w}'$											

3. **Linear Space Alignment** [10 points]

Consider two sequences $\mathbf{v}$ = TG and $\mathbf{w}$ = ATCG of length $m = |\mathbf{v}| = 2$ and $n = |\mathbf{w}| = 4$, respectively. In this exercise, we will compute an optimal global alignment of the two sequences using the Hirschberg algorithm. We will use a score of +1 for a match, -1 for a mismatch, and -1 for a insertion/deletion (i.e. a gap penalty of 1).

a. The initial call is HIRSCHBERG$(0, 0, m = 2, n = 4)$. We need to identify the middle vertex $(i^*, n/2 = 2)$. **(i) Fill out** the following table for this initial call and **(ii) indicate** i^*. [2 points]

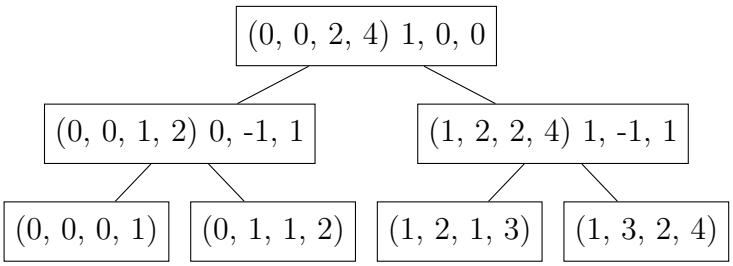
i	prefix(i)	suffix(i)	wt(i)
0	-2	0	-2
1	0	0	0
2	-1	-2	-3

Middle vertex i^*: 1

b. What are the two recursive calls that are made in this initial invocation HIRSCHBERG$(0, 0, m, n)$? [1 point]

HIRSCHBERG(0, 0, 1, 2) and HIRSCHBERG(1, 2, 2, 4)

c. **(i) Give the recursion tree**, where each vertex corresponds to an invocation of HIRSCHBERG. See Lecture 1 for an example of a recursion tree. **(ii) Label each vertex** of this tree by the used arguments (i, j, i', j'). In addition, **(iii) label each *internal* vertex** by the value of i^*, prefix(i^*) and suffix(i^*). Make sure to include the bases cases as leaves of the tree. [5 points]



d. (i) Indicate the reported vertices in the table and (ii) give the final alignment. [2 points]

Reported vertices ('X'):

	0	A	T	C	G
0	X	X			
T			X	X	
G					X

Final alignment:

-	T	-	G		
A	T	C	G		

4. **BLOSUM** [10 points]

   Consider the following four blocks on the alphabet $\Sigma = \{A, T, C, G\}$.

ATCGA	TTC	AAAA	AA
ATCGA	TTC	GTTT	AT
TTCGA	TCC	TAAA	AG
AACGA	CTC	AGTA	AC
CTAGA	TTG		AA
AACAA			AG

   Using $L = 0$, such that the above four blocks are not pruned down, compute the BLOSUM0 scoring matrix. Use $\lambda = 0.5$, the natural logarithm and round up to the nearest integer (i.e. take the ceiling). **(i) Give q_x and (ii) $p_{x,y}$ for each $x, y \in \Sigma$. Clearly indicate the denominator used for computing these two quantities.**

(i) q_x **(denominator** $N = 73$**):**

$$q_A = \frac{31}{73}, q_T = \frac{19}{73}, q_C = \frac{13}{73}, q_G = \frac{10}{73}$$

(ii) $p_{x,y}$ **(denominator** $N_{\textbf{align}} = 159$**):**

$$p_{A,A} = \frac{44}{159}, p_{T,T} = \frac{19}{159}, p_{C,C} = \frac{16}{159}, p_{G,G} = \frac{11}{159}, p_{A,T} = \frac{25}{159},$$
$$p_{A,C} = \frac{11}{159}, p_{A,G} = \frac{13}{159}, p_{T,C} = \frac{10}{159}, p_{T,G} = \frac{4}{159}, p_{C,G} = \frac{6}{159}$$

BLOSUM0 Scoring Matrix:

$$s(x,y) = \left\lceil \frac{1}{\lambda} \ln\left(\frac{p_{x,y}}{q_x q_y}\right) \right\rceil, \quad \lambda = 0.5$$

.

$$\begin{array}{r|l}
s(A,A) & 2\ln(\frac{44/159}{0.4247 \times 0.4247}) = 0.85646 \\
s(T,T) & 2\ln(\frac{19/159}{0.2603 \times 0.2603}) = 1.13515 \\
s(C,C) & 2\ln(\frac{16/159}{0.1781 \times 0.1781}) = 2.30941 \\
s(G,G) & 2\ln(\frac{11/159}{0.1370 \times 0.1370}) = 2.60948 \\
s(A,T) & 2\ln(\frac{25/159}{0.4247 \times 0.2603}) = 0.70493 \\
s(A,C) & 2\ln(\frac{11/159}{0.4247 \times 0.1781}) = -0.17805 \\
s(A,G) & 2\ln(\frac{13/159}{0.4247 \times 0.1370}) = 0.68078 \\
s(T,C) & 2\ln(\frac{10/159}{0.2603 \times 0.1781}) = 0.61042 \\
s(T,G) & 2\ln(\frac{4/159}{0.2603 \times 0.1370}) = -0.69743 \\
s(C,G) & 2\ln(\frac{6/159}{0.1781 \times 0.1370}) = 0.87248
\end{array}$$

.

$s(x,y)$	A	T	C	G
A	1	1	0	1
T	1	2	1	0
C	0	1	3	1
G	1	0	1	3

5. **Bonus: Total Number of Global Alignments** [6 points]

   In this bonus question, we are going to determine the total number of *global alignments* that exist given two strings $\mathbf{v} \in \Sigma^m$ and $\mathbf{w} \in \Sigma^n$. We will assume without loss of generality that $m \leq n$. Recall the matrix representation of an alignment. This is a $2 \times k$ matrix where $k \in \{\max\{m, n\}, \ldots, m + n\}$ such that there is no column with two gaps. Thus, the number k of columns varies from $\max\{m, n\}$ to $m + n$.

   a. Explain why, in general (i.e. no prior knowledge on how m and n are related), the number k of columns varies from $\max\{m, n\}$ to $m + n$. [1 point]

   .

   b. Explain why $k \in \{n, \ldots, m + n\}$ for the case where $m \leq n$. [1 point]

   .

   c. Suppose that the alignment has length $k \geq n$. In how many different ways can we insert $k - n$ gaps in the second sequence $\mathbf{w}$, yielding gapped sequence $\mathbf{w}'$? [1 point]

   Hint: Observe that $\mathbf{w}'$ has length k.

   .

   d. Let $\mathbf{w}'$ be a gapped sequence of length $k \in \{n, \ldots, m + n\}$ such that removing the gaps yields the original sequence $\mathbf{w}$. In how many ways can we insert gaps in $\mathbf{v}$ to obtain an alignment with $\mathbf{w}'$ of length k? [1 point]

   Hint: Recall that an alignment does not contain columns with two gaps. In how many different ways can we insert gaps in $\mathbf{v}$ subject to this condition?

   .

e. How many alignments of $\mathbf{v}$ and $\mathbf{w}$ are there of a given length $k \in \{n, \ldots, m+n\}$? How many alignments are there of any length? [1 point]

Hint: Combine your answers to the previous two questions.

.

f. Give an example of a scoring function $\delta : (\Sigma \cup \{-\}) \times (\Sigma \cup \{-\}) \to \mathbb{R}$ such that the number of optimal global alignments equals your answer to the previous question. [1 point]

Hint: Think of a border case.

.