Key

Midterm Exam

CS 4364/5364

Spring 2021

Instructions: Please read all of the instructions below before you begin:

- Read all of the questions in the exam before you begin.
- Questions marked with a dagger (†) are required for students in CS 5364, and bonus (optional) for those in CS 4364
- Figures (pictograms) can be included if they help describe a solution, and are encouraged if they are clear.
- Remember that unanswered questions will receive 0 credit, any reasonable try at a response will receive at least half-credit. If you feel you're unable to provide a reasonable answer to a question, you can answer with "I cannot provide a reasonable attempt for this question", which will be provided quarter-credit.
- Students are allowed to bring any written material with them to the exam, but no electronic devices can be used during the exam.
- if you need extra room, please use the *back of the previous page* rather than the current one (when it is folded open your whole answer should be showing).

1. Use the *Needleman-Wunch* dynamic programming table for S = TTTATG and T = TCTAT below to the next questions.

		T	T	T	A	T	G
	0	←-1	←-2	←-3	←-4	←-5	←-6
T	↑-1	↖8	↖7	↖6	←5	↖4	←3
C	↑-2	↑7	↑6	↑5	↑4	↑3	↑2
T	↑-3	↑6	↖15	↖14	←13	↖12	←11
A	↑-4	↑5	↑14	↑13	↖22	←21	←20
T	↑-5	↑4	↑13	↖22	↑21	↖30	←29

(a) How many co-optimal alignments of the two strings are there? What are they?

(b) What is the optimal alignment of $S[1....3]$ and $T[1...5]$? (note these are prefixes TTT and TCTAT)

(c) What is the indel penalty used to construct the table? match score?

(d) †Can you determine the mismatch penalty from the table above? **Explain why or why not.**

(a) 4 6pt 1pt ea.

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

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

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

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

(b) T - T - T
T C T A T

(c) indel = -1
match = 8

(d) no, no mismatches in table

2. Below is an algorithm description for a given problem (not defined on purpose).

   - assume you are given a string $S = s_1 s_2 s_3 ... s_n$.
   - set $S^R = s_n s_{n-1} s_{n-3} ... s_1$, that is S^R is the reverse of S.
   - construct a new string $T = S\$S^R$, where $\$ \notin \Sigma$.
   - compute the maximum prefix overlap, $M_i(T)$ for each $2 \leq i \leq (2n+1)$.
   - return $M_{n+2}(T) == n$.

   (a) What is returned when (i)$S =$ `ABCBA`? (ii)$S =$ `ABCDBA`?

   (b) What is the running time of the algorithm (in terms of n, the length of S)?

   (c) What does this algorithm do?

(a) i) True

ii) False

(b) o(n)

(c) finds if S is a palindrome.

3. Given the alignments below, determine the number of matches, mismatches, gaps, and indels; place the counts in the table.

A_1	GCT-AC-TTGC -CTG-CA-TGC
A_2	GCTACTTGC -CTGCATGC
A_3	GCT--ACTTGC -CTGCA--TGC

Matches mt	Mismatches ms	Gaps gp	Indels id
6	0	5	5
6	2	1	1
6	0	3	5

†Determine at set of values for α, β, γ, and δ such that each of the alignments A_3 is optimal under the following scoring:

$$\alpha\text{mt} + \beta\text{ms} + \gamma\text{gp} + \delta\text{id}$$

assuming A_1, A_2, and A_3 are the only possible alignments of the two sequences.

α	β	γ	δ
10	-10	-3	-1

4. Use the generalized suffix tree below to answer the next questions.

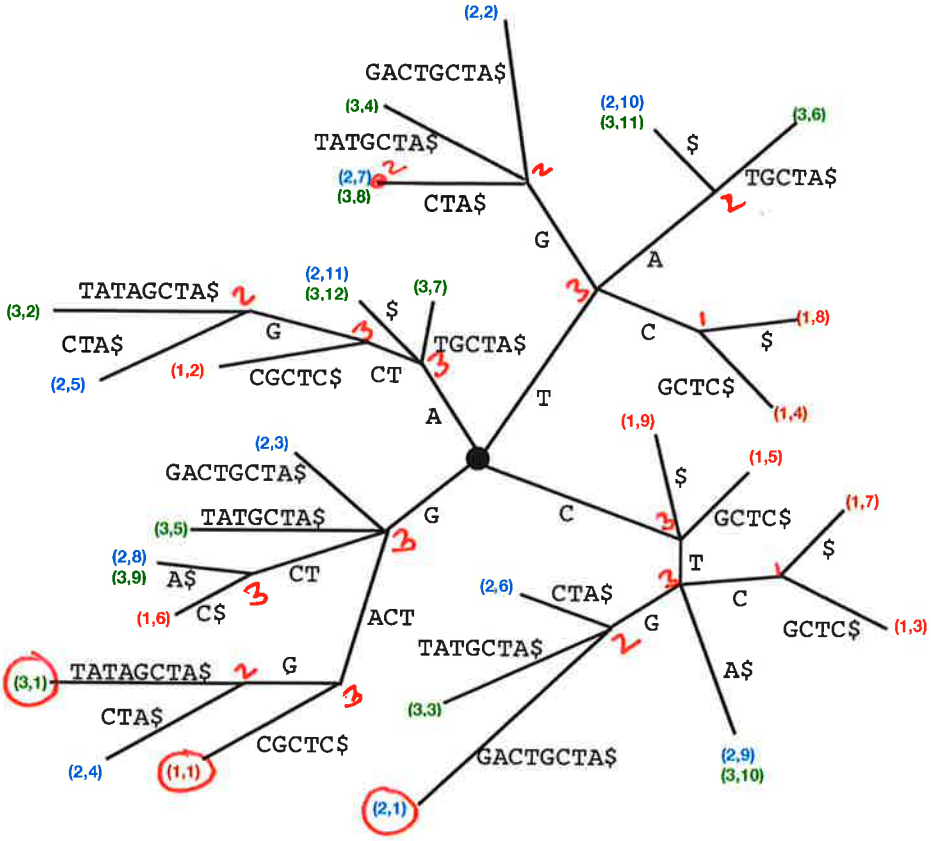

Figure 1: Suffix tree for question 4

(a) What are the 3 strings encoded in the tree?

(b) What is the longest string contained in **at least two** strings?

(c) Name two possible alphabets that could be used to produce the strings used to construct the strings?

(a) G A C T C G C T C
C T G G A C T G C T A
G A C T G T A T A G C T A

(b) G A C T G T G C T A shared leaf.

(c) Examples
DNA
Capital letters

5. †True or False: For global pairwise alignment of two strings of size m and n

$$2\text{mt} + 2\text{ms} + \text{id} + \text{gp} = (m + n).$$

Where mt is the count of matches, ms is the count of mismatches, and id is the number of indels, and gp is the number of gaps. **Explain your answer.**

False

2 mt + 2 ms + id = (m+n) (no gaps)

therefore cannot be true.

6. Given two sequences S and T (not necessarily the same length), let G and L be the scores of an optimal global alignment and an optimal local alignment, respectively.

   (a) True or false — $\forall S, T \in \Sigma^* : L \geq G$. **Explain your answer.**

True;

Local looks at best substrings and is global is a restriction looks at 1 substrg.