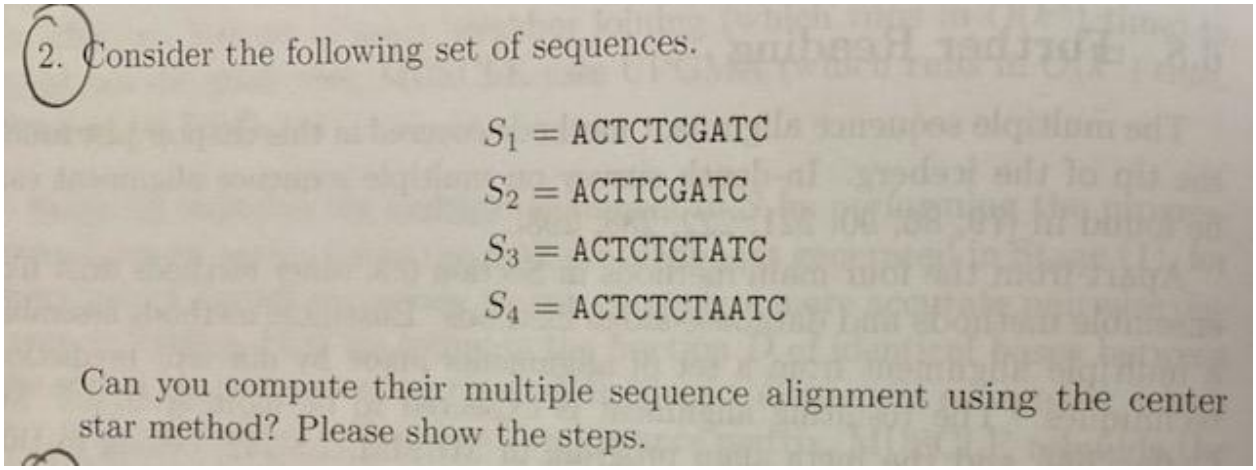
CSCI 451/551 Fall 2023 Homework 5
Due: Nov 28, at the start of class

(Group submissions: please make sure all group members are listed)

Problem 1 (10 pts) Exercise 2 (textbook section 6.9, pg 152)

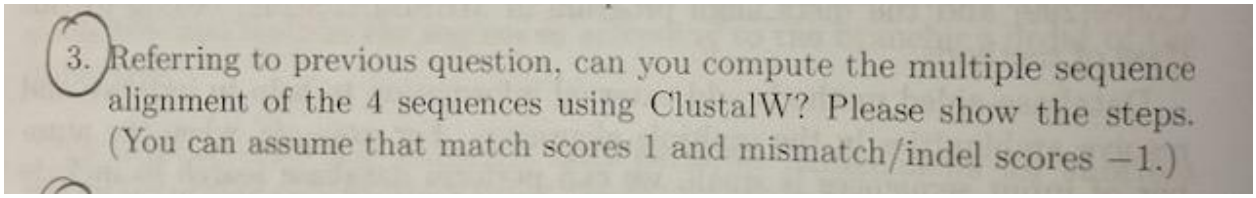


2. Consider the following set of sequences.

$S_1 = \texttt{ACTCTCGATC}$
$S_2 = \texttt{ACTTCGATC}$
$S_3 = \texttt{ACTCTCTATC}$
$S_4 = \texttt{ACTCTCTAATC}$

Can you compute their multiple sequence alignment using the center star method? Please show the steps.

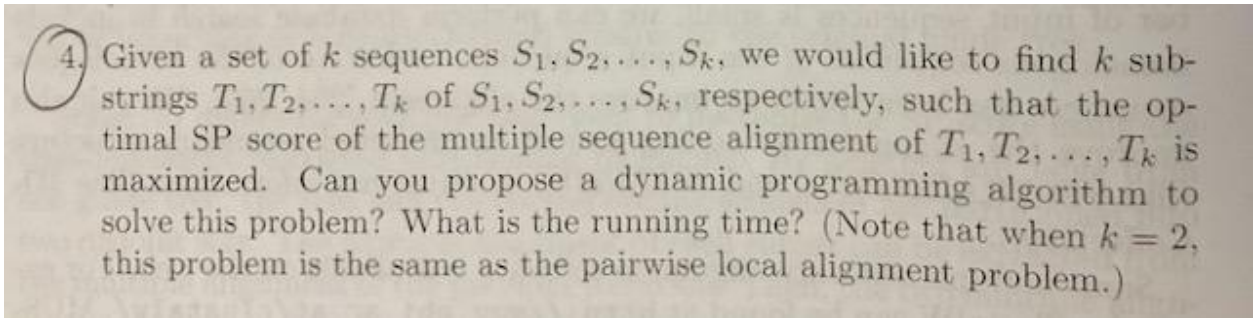
Problem 2 (5 pts) Exercise 3 (textbook section 6.9, pg 152)



3. Referring to previous question, can you compute the multiple sequence alignment of the 4 sequences using ClustalW? Please show the steps. (You can assume that match scores 1 and mismatch/indel scores -1.)

There are a few online CLUSTALW servers, one is: https://www.genome.jp/tools-bin/clustalw

Problem 3 (10 pts) Exercise 4 (textbook section 6.9, pg 152)



4. Given a set of k sequences $S_1, S_2, \ldots, S_k$, we would like to find k substrings $T_1, T_2, \ldots, T_k$ of $S_1, S_2, \ldots, S_k$, respectively, such that the optimal SP score of the multiple sequence alignment of $T_1, T_2, \ldots, T_k$ is maximized. Can you propose a dynamic programming algorithm to solve this problem? What is the running time? (Note that when $k = 2$, this problem is the same as the pairwise local alignment problem.)

Hint: what other dynamic programming problem is this similar to? You can just explain briefly what is needed to make it work for this problem.

Problem 4 (10 pts) *Fitch's algorithm:* For the following five sequences, first build a binary tree where the sequences are the leaves (you can just guess the tree topology that you think leads to a good score). Then apply Fitch's algorithm to predict the ancestral sequences at each internal node in the tree.

S_1 = acctt S_2 = tcggc S_3 = tactt S_4 = atcgt S_5 = acata

Problem 5 (20 pts) Programming project: Pick <u>one</u> of the following programming projects:

Project A: *Center Sequence Finder*

Implement the first two steps of the *Center Star Algorithm*. Your program should take as input a FASTA file specifying a set of sequences, as well as two cost function parameters, alpha and beta, where delta(x,y) = alpha > 0 and delta(x,-) = beta > 0 (Note: delta(x,x) = 0).

The program should output the center sequence, based on the specified distance function parameters.

Demonstrate the algorithm works on several examples, including the set of sequences from Problem 1. For one example, also complete the next step of the algorithm to find the MSA for the sequences (this can be done by hand, or, you can also do the bonus question below).

Bonus (5 pts) Implement the entire Center Star Algorithm so that your program also finds and prints out the MSA.

Project B: Fast LCS for Permutations

Implement the faster LCS algorithm we discussed (that has O(n lg n) running time). Your program should create two random permutations of length n (n should be supplied as an input parameter to the program). For example, if n = 10, it might create:

```
2 3 1 4 5 7 6 8 10 9
9 10 7 4 5 6 8 1 2 3
```

The program should then find the LCS and output the original permutations, as well as the LCS. Demonstrate your program on a couple sample inputs (one with length 200 or greater).

Also explain, <u>carefully</u>, why your implementation achieves O(n lg n) time.

Bonus (5 pts): Also implement the standard table-based $O(n^2)$ algorithm and collect running times for various sizes of n and plot the running times of the two algorithms.

Please also have your 451/551 paper/project decided by the due date.