605.420 Foundations of Algorithms Project 3

Design and implement algorithms to solve the longest common subsequence problem (chapter 15.4).

S1 = ACCGGTCGACTGCGCGGAAGCCGGCCGAA
S2 = GTCGTTCGGAATGCCGTTGCTCTGTAAA
S3 = ATTGCATTGCATGGGCGCGATGCATTTGGTTAATTCCTCG
S4 = CTTGCTTAAATGTGCA

Compare each of the provided strings to each other and to your test cases. Compare by pairs only, finding the LCS of the pair. Gaps in matching the substring are allowed.

In the writeup be sure to consider your experiences with the problem and also consider the efficiency. Why is it useful and relevant to Bioinformatics? Don't forget to review the programming assignments guidelines.

Here is a related YouTube video you might interesting "A Dynamic Programming Algorithm - Sequence Alignment " - By Tim Roughgarden https://youtu.be/xccdfMM6l7c. Roughgarden has other useful videos as well.

Count individual comparisons to use as a basis for cost efficiency in terms of the string lengths. If you want to try timing, please do it in addition to, not instead of, counting comparisons.

A file with required input is provided. The required input is used to demonstrate correctness. You should supplement this with your own cases that demonstrate correctness in various standard error situations,

To demonstrate the asymptotic cost you will need to create, additional input sets with larger strings. You need to collect enough data to have a meaningful comparison of the theoretical efficiency to the observed efficiency. The analysis should include comments about what you learned, what you might do differently next time, justification of your design decision, and issues of efficiency with respect to time and space. Your analysis must include a table and a graph of the asymptotic costs that you observed. Be sure to consider your experiences with the problem and particularly consider the efficiency. Why is it useful and relevant to Bioinformatics? **Before you hand this in, be sure to reread the Programming assignments guidelines.**