CSE 3500 — Algorithms and Complexity — Yufeng Wu — Spring 2016
Programming Assignment II: Dynamic Programming ——- Due: 04/13/16 at the end of the day.

In this assignment, you are going to implement the dynamic programming algorithm for finding the longest common subsequence (LCS), and also analyze some data using your implementation. First, you need to implement the LCS algorithm using your favorite language (perhaps Java). Then run your program on the provided data files to examine how your program behaves on various types of data. Then you will put your code into action to analyze some datasets. These analyses will use the LCS as a tool in comparing sequences. Here are more detailed information on the provided data. After de-compression, you should see these types of files.

Type-I There are files with names like listSeqs-errorxxx-lxxx.txt, which are generated with different error rates. Each file contains 10 sequences (each line for a sequence). These ten sequences are generated from a single consensus sequence (its length is specified in the filename) by adding some modifications (errors including changing of the characters, insertion of a new character or deleting a character in the consensus). I generated two types of sequences: those with low errors (i.e. these sequences match more the consensus sequences and should have longer LCS) and those with high errors (these sequences tend to look more different and thus should have shorter LCS).

Type-II There are files with names like listSeqs-consensustest-errorxxx-lxxx.nsol.txt. Each file contains 10 sequences generated from one of the two different types of consensus sequences: one with four letters A, T, C, G and the other with only two letters. Each sequence randomly picks one of the two consensus sequences (again with some errors introduced). There can be low or high level of errors from consensus sequences.

Implementation

Your program needs to both find the length of the LCS and also the LCS itself. Recall in the class I described how to use traceback to find the LCS from the dynamic programming table.

Analysis

Run your program to find the LCS of each pair of the 10 sequences stored in a single file (i.e. you only need to compare sequences that are contained in the same file). You should perform the following analysis. Note: You don't need to show results for all lengths. You can just show results that you deem to be important for your observations.

Time First report the running time for computing the LCS of a pair of sequences of certain length (for type-I files).

Error rate vs. LCS Collect the results on the normalized length of the LCS (i.e. the length of LCS divided by the total average sequence lengths of the two involved sequences) for each file of type-I. Then plot a graph of the normalized lengths for the two types of error rates. Can you draw some conclusion about the correlation of the normalized LCS length and error rates?

Consensus type Find the LCS between each pair of sequences on the files with sequences generated from two different types of consensus (type-II files) sequences. Plot the normalized LCS length of these pairs with various sequence length in a graph. Can you draw some conclusion on the kind of sequences you have based on such analysis (when compared with the results from sequences coming from a single consensus)? Moreover, can you find out (by examining the found LCS) the two characters that are in the consensus sequences? Do error rate and sequence length matter in finding these two characters? You may also compare the results with that of the sequences with a single consensus: can you confidently tell these two cases (i.e. sequences generated by two different types of consensus vs. those generated by a single consensus) apart by checking the LCS lengths?

What to submit?

Submit a short report containing the following.

Settings Write down the language you use, the machine (its CPU frequency and memory size) you use for testing your program.

Results Present the running time analysis on various sizes and types of sequences. How well does your program scale with the input size? Show analysis results on the relationship of error rates and LCS length, and the effect of consensus sequences.

Conclusion Tell me what you learned by doing this assignment.

Code Attach the source code of your implementation. If it is short enough, you may simply include your code as part of your report.