Bhavyansh Mishra

Data Structures and Algorithms II

Project 5

Functional Decomposition

Longest Common Subsequence

*(Using Dynamic Programming)*

**Part 1**

**main.c**

Display the original strings and the longest common subsequence of characters.

**lcs.c**

Read a file

Containing a pair of character strings ("twoSequences.txt") corresponding to the sequences,

Compute the LCS

Utilize the entire "c" array to reconstruct the sequence itself

(If you use the "b" array, it will cost you 10%).

**Part 2**

(Will not make any attempt to recount the actual subsequence).

**similarity.c**

Will calculate measures of similarity among an arbitrary number of character strings

In a file named "multipleSequences.txt".

This file will start with an integer (on its own line in the file) indicating the number of strings that follow.

That number of strings will follow, one per line.

Utilize an approach of keeping only the 2xM entries needed to compute the maximum value of EACH LCS.

Produce as output, a table that will look like this:

    1  2  3  4  5  6  7

 1  -  H  M  D  M  L  D

 2  -  -  H  H  D  D  H

 3  -  -  -  L  M  D  M

 4  -  -  -  -  M  L  M

 5  -  -  -  -  -  M  L

 6  -  -  -  -  -  -  M

 7  -  -  -  -  -  -  -

Obviously, the 1, 2, 3, ... are labels for the strings, and the upper triangle holds a measure of similarity between pairs of strings {1,2}, {1,3}, {1,4}...

The possible entries in the table are:

H = high similarity between the strings

M = medium similarity between the strings

L = low similarity between the strings

D = the two strings are dissimilar