CSCI 451 Homework 2

Samuel Shissler, Brendan Tracey, and John Trapp

September 26, 2017

Smith-Waterman Lab

Test Log

Please view the text file containing the test logs ('_' is a special character in LaTeX).

Exercises

Exercise 2

The Longest common subsequence (LCS) can be solved using dynamic programming. Given two strings $S1$ and $S2$ we can make a matrix m with columns i and j. The algorithm to find the LCS would look like this: If the value $S1(i)$ is equal to the value at $S2(j)$ then the alignment score at $m(i, j)$ is equal to the score at $m(i-1, j-1)+1$. Otherwise the alignment score at $m(i, j)$ is equal to the alignment score at $m(i-1, j-1)$.

Exercise 9

The algorithm to find the minimum number of operations to transform string $S1$ into string $S2$ using only the given functions 1) Insert 2) Delete 3) Replacement 4) Reversal of a substring would like something like this.
On matrix m calculate all the optimal alignment score between $S1$ and $S2$ using the Smith-Waterman algorithm scoring only for insert, match, delete: Calculate step 1 on all possible reversals of $S1$ (reversing a substring of $S1$ at position a through b has the same effect on the alignment score as reversing a substring at position a through b of $S2$. The time complexity would be $O(n^3)$, because the number of substring reversals that can be done on a string of length n is $(n)x(n-1)$ which is $O(n^2)$ time complexity, and the Smith-Waterman Algorithm can be reduced to linear time. However each n^2 instance of the substring reversal must execute the Smith Waterman Algorithm, meaning the time complexity is $O(n^3)$.