Justin-Anthony Joco
ECE 368

PA05 Report

The purpose of this assignment was to design a program that finds the longest conserved gene sequence among a set of species. Since this problem cannot be solved simply through arrays, I attempted to solve it through graphs.

Given the first two integers of the binary file are the length of the sequence and the number of species, respectively, Firstly, I created a structure, “Graph”, which stores the number of integers in the graph, a list of all the integers in the sequence, and the adjacency matrix. I read the first sequence into graph->vertex_array, then I stored the adjacency matrix into graph->edge_matrix. For every sequence after the first, I save the adjacency matrix into “temp_matrix.” I AND every element in both matrices and save the result into ret_matrix. After all sequences’ matrices have been ANDed to ret_matrix, I conduct depth first search on ret_matrix in order to find the longest conserved sequence by using a counter for every tree in the forest. The tree with the highest counter will be denoted as the max_counter. The conserved sequence would be saved into an int array, but I did not implement this. However, I was able to return the size of that array into “size_of_seq,” which would be printed to stdout.

Overall, the algorithm’s time complexity is O(n^2*m). Indeed, the time complexity of DFS is also O(n^2) as opposed to O(n+E). Even though the program requires O(n^2) memory space, the time complexity still remains high at about O(n^2*m). This is all due to the use of adjacency matrices instead of lists. Unfortunately, my algorithm is greedy since it does not find the optimal length of the genome sequences, but finds a length close enough.