Programming Assignment 5 Report Praveen Seeniraj

My algorithm takes two elements and checks their relative positions in each of the arrays. If the two numbers lie one after the other in all of the arrays that are in the input file. If so, it adds those 2 elements to a graph. Once the entire processes, there is a now a graph with multiple power sets of size 2. Each of those of those power sets are linked with other power sets. Since the arrays are permutations of each other, the properties of the power sets are transitive. Hence the easiest way to solve to the problem is to find the longest chain the graph. Since this is a weighted directed graph the easiest way to find the longest path was to create a variation of topological sort. Each of the nodes are weighted and are recursively used to traverse the adjacency list of the graph and then the graph nodes of the adjacency list. While recursively increasing how deep the node is located from the source node.

The graph creation part of the algorithm is the most inefficient part of the algorithm with a time complexity of O(n^2).

The algorithm that calculates the longest chain is of the time complexity O(n + E) E is the number of edges in the graph.