Kimberly Arenas

Lab 7

Introduction:

Dynamic Programming is a method for solving a complex problem by breaking it down into a collection of simpler subproblems, solving each of those subproblems just once, and storing their solutions using a memory-based data structure. Edit-distance is a dynamic programming way to quantify how dissimilar two strings are by keeping count of how many operations it will need for both strings to be the same. In this lab we will implement the algorithm to perform Edit-distance with some testing files.

Proposed solution design and implementation:

To start of the edit-distance algorithm we must commence by creating two basic base cases to check if either string is empty: if the length of the first string is zero, return the length of the second string, vice versa on the second string. Then the fist element of each string will be compared and once they are not similar the operations of insert, remove and replace begin. When the process above is completed it will take count of how many of the three operations were needed for the two strings to become identical.

The three different test cases that will test out the algorithm will be:

1. Running it on two distinct strings.
2. Running it on the same strings.
3. Running it on one string and an empty string.

Lastly the program will have an additional section after the test cases are completed and it will be for the user to test two strings of their own!

A screenshot of a cell phone

Description automatically generatedExperiment Results:

Conclusion: I now have a better understanding of how edit-distance works in the coding world! If I were to do this better I would give the user more opportunities to input limited amount of words they would like to know their distance.

I certify that this project is entirely my own work. I wrote, debugged, and tested the code being presented, performed the experiments, and wrote the report. I also certify that I did not share my code or report or provide inappropriate assistance to any student in the class.