

Kennesaw State University
Department of Computer Science
CS4306: Algorithm Design and Analysis
Spring 2024

Bonus Assignment 3 (Due: May 4th, 2024) Instructor: M. Alexiou

The Levenshtein Distance Problem:

The Levenshtein distance is a way of quantifying how different two strings are from one another by counting the minimum number of operations required to transform one string into the other. More specifically, the distance between two words is computed as the minimum number of single-character edits (i.e., insertions, deletions, or substitutions) required to change one word into the other. Each of these operations has a unit cost.

Example:

For words house → horse: There is a substitution of r for u which indicates distance of 1 (1 edit required).

For words horse → morse: There is a substitution of m for h which indicates distance of 1 (1 edit required).

For words house → horses: There is a substitution of r for u and the insertion of s at end which indicates distance of 2 (2 edits required).

You are asked to write a program that measured the Levenshtein distance between two words using a Dynamic Programming approach:

1. **Input:** Absolute paths to the input and output files
2. **Output:** The distance measured for each pair of words.

Example of Input/Output Format in text Files:

To streamline the inputs and outputs for grading purposes, please format your input text files as comma-delimited lines as follows:

Input File:

word_11, word_12

word_21, word_22

...

Similarly, the output files should also be formatted as follows::

Output File:

word_11, word_12, distance_1

word_21, word_22, distance_2

...

Implementation Instructions:

(a) Please provide at least 10 different word pairs and their corresponding outputs to demonstrate your understanding of the algorithm and to test your system on edge cases. You should put your test data in a folder and name it “InputsOutputs”. Name your test inputs as “Input.txt”, “Ouput.txt”, etc.

(b) Please provide detailed comments to carefully explain your code and the functions that you have developed.

(c) In your program, include the Bellman-equations that you have identified for your algorithmic solution as a comment.

(d) Upload one zip archive per team (with team member names and email addresses) that contains text files (e.g., README, sample inputs and outputs), and source code file(s) to HW 3 on Assignments on D2L by May 4th, 2024. **(Only one submission per team!)**. Your assignment will be graded on correctness based on the test inputs that both your team and I will provide as well as on the well-documented source code and the efficiency of your algorithm. Late submissions will be penalized at the rate of 25% / day and no submissions will be accepted after May 5th on D2L.