Levenshtein String Matching

CSCI 5870

Goal

Implement Levenshtein's algorithm for finding the minimum distance between two strings.

Details

This project has three main goals:

1. It must correctly implement Levenshtein's algorithm
2. It must use space efficiently (i.e., $\Theta(\min(|s_1|, |s_2|))$ space)
3. It must work with an arbitrarily long strings

Data format

The strings are contained in two files, whose names are given on the command line.

Output

The program should output two values: the minimum distance d_{min}, and a "similarity coefficient" $s = 1 - \frac{d_{min}}{\max(|s_1|,|s_2|)}$. The similarity coefficient is 1 if the strings are identical, and 0 if the minimum distance must replace, delete or insert every character.

Test runs

Generate several (at least ten) test cases, with strings between 10 and 10,000 characters (one case must have 10, one must have 10,000). In each test case, the lengths of the strings do not need to match. Test cases should include very similar string pairs, and very dissimilar string pairs.

What to turn in

Turn in an electronic copy of your code, and the results of your test runs.