Gorilla Report

Author 1 and Author 2

May 19, 2020

1 Results

The script shows all of the results to be correct. The program takes a very short time to run; the biggest input runs in about a second, with the input reading being only about 50ms of the total time.

2 Implementation details

To implement the solution a matrix storing all the optimal routes was created for each string alignment. This matrix was created using an iterative solution where we store a different value depending on if we insert a blank in one of the strings or not. We then read through the matrix, starting at the end and depending on the value at a given coordinate, we can tell if we need to insert a blank space in one of the words or not. We also use a lookup table for converting the characters in the alphabet to integers, so we can use them as indices in the given cost matrix.

The algorithm starts by building the optimum matrix, which is of size m*n, where m and n are the string lengths of the respective strings. The whole matrix has to be built and thus this is a O(m*n) operation. Recreating the words with blanks takes m + n steps, where each step takes constant time, and recreating the words is thus a O(m + n) operation. The total time complexity is thus O(m*n + m + n) = O(m * n).