EECS 325 – Winter 2023
Analysis of Algorithms

Homework 5

Instructions:

- This homework is due by 11:59PM Pacific Time on **Thursday, March 9th**. You will receive one grace period allowing you to submit a homework assignment up to two days late for the quarter. Subsequent assignments submitted up to two days late will incur a 20% penalty. Assignments submitted more than two days late will not be graded and will receive a score of 0. Your lowest homework score of the quarter (after applying late penalties) will be dropped.
- You must submit your written solutions as a single .pdf file on Canvas with your name at the top. Typesetting your written solutions using LaTeX is strongly encouraged. You must write your programming solutions as a single .py file. We have provided a skeleton file.
- You are welcome and encouraged to discuss the problems in groups of up to three people, but **you must write up and submit your own solutions and code**. You must also write the names of everyone in your group on the top of your submission. Students in the honors and non-honors section may collaborate.
- The primary resources for this class are the lectures, lecture slides, the CLRS and Erickson algorithms textbooks, the teaching staff, your (up to two) collaborators, and the class Piazza forum. We strongly encourage you only to use these resources. If you do use another resource, make sure to cite it and explain why you needed it.
- There are three questions, worth a total of 30 points.
- You must justify all of your answers unless specifically stated otherwise.

Questions

Question 1 (Variable-cost edit distance, 10 points)**.** In the *variable-cost edit distance* problem, your goal is to determine the *minimum total cost* of transforming a string $S_1[1, \ldots, n_1]$ into a string $S_2[1, \ldots, n_2]$ via a sequence of insertions, deletions, and mutations when these operations have potentially different costs $c_i > 0$, $c_d > 0$, and $c_m > 0$, respectively. The "normal" edit distance problem corresponds to the case where $c_i = c_d = c_m = 1$.

Modify the tabulation-based dynamic programming algorithm that we saw in class for computing the generalized edit distance between strings S_1 and S_2 for given values of c_i, c_d, and c_m. Implement your algorithm as the `edit_distance` function in the skeleton code.

Question 2 (2 and 3-string LCS, 14 points)**.**

a. (5 points.) Implement the dynamic programming algorithm for computing the length of the longest common subsequence (LCS) of two strings that we saw in class as the `lcs` function in the skeleton code.

b. (9 points.) Give a dynamic programming algorithm for computing the length of the longest common subsequence of three strings $S_1[1, \ldots, n_1]$, $S_2[1, \ldots, n_2]$, and $S_3[1, \ldots, n_3]$. Implement your algorithm as the `lcs3` function in the skeleton code.

For example, if S_1 = `'AGGCA'`, S_2 = `'CTTGA'`, and S_3 = `'GTA'`, then the length of the longest common subsequence between S_1, S_2, S_3 is 2 because each string contains `G`, `A` as a subsequence (and $|S_3| = 3$, but `G`, `T`, `A` is not a subsequence of A or B): S_1 = `'AGGCA'`, S_2 = `'CTTGA'`, and S_3 = `'GTA'`.

(**Hint**: Your algorithm should use a table L with three indices: $L[i, j, k]$.)

Question 3 (Measuring the evolution of the COVID genome, 6 points.)**.** Compute (1) the edit distance d between and (2) the length k of the longest common subsequence (LCS) in the National Institutes of Health's (NIH's) reference COVID genome and the provided COVID Omicron BA.1 variant genome.[1] These are each provided in the text files `COVID-RefDec19.txt` and `COVID-OmicronBA1.txt`. Only consider the *letters* in these files, and not numbers, punctuation, or white space. (A parser that extracts just the letters is provided in the skeleton code.) For edit distance, use $c_i = c_d = c_m = 1$. For the length of the LCS, you should use the `lcs` function and not the `lcs3` function.

Your answer to this question should be the pair of numbers output by your program `my_homework.py` when running it as follows:

```
python my_homework.py COVID-RefDec19.txt COVID-OmicronBA1.txt
```

(**Hint**: This question is just testing your solutions to the previous questions.)

[1] These are available at https://www.ncbi.nlm.nih.gov/nuccore/NC_045512.2 and https://www.ncbi.nlm.nih.gov/nuccore/OX315743.1, respectively. The former was sequenced in late December 2019/early January 2020 from one of the first patients who fell ill.