

Assignment 2

Due: Monday, 19 October 2020, 3:30 p.m. (before the start of the lecture)

To submit the assignment, upload the following three files to Blackboard before the submission deadline (replace “99999999Z” below by your student ID number):

- A source code file named “LCS99999999Z” plus a filename extension, e.g., “LCS99999999Z.c”, for the program in question 1. Include your name and student ID number in the source code.
- A .pdf-, .doc-, .docx-, .txt-, or .jpg-file containing a report with your solutions to questions 2–6. The filename should start with “A2_99999999Z”. Include your name and student ID number at the top of each page. Scanned, handwritten solutions are OK as long as they are easily readable.
- A file named “Max99999999Z.txt” with the extreme example you found in question 3.

The maximum possible score is 15 marks. Late submissions will not be accepted. The assignment is to be completed individually, i.e., every student must turn in his or her own solutions. Students are allowed to discuss solution strategies with other students but they are not allowed to copy each others’ answers.

1. Write a Java, Python, C, or C++ program that outputs a longest common subsequence of three input sequences X , Y , and Z of symbols from a fixed alphabet. For example, if the input is:

$X = \langle N, I, N, E \rangle$,

$Y = \langle S, I, N, G, I, N, G \rangle$,

$Z = \langle N, I, N, J, A, S \rangle$

then the output should be $\langle N, I, N \rangle$. For this input, the optimal solution is unique. For inputs where the optimal solution is *not* unique, your program may output any longest common subsequence. (If you want to use another programming language, get the instructors’ approval first.) (4 marks)

2. In the report, briefly explain how your program works and analyze its time and space complexity. Also, show a small example input created by you and the corresponding output. (2 marks)
3. When all three input sequences have the same length n , what is the largest n for which your program manages to compute a solution in less than one minute? Is the execution time or running out of memory the main issue here? Put this “extreme” example in “Max99999999Z.txt”. (1 mark)
4. A *binary sequence* is a sequence over the alphabet $\{0, 1\}$. For example, $\langle 0, 1, 0, 1, 1, 0, 0, 0 \rangle$ is a binary sequence. Give an example of three binary sequences X , Y , Z , each of length 8, such that the length of a longest common subsequence of X and W , where W is a longest common subsequence of Y and Z , is different from the length of a longest common subsequence of X , Y , Z . You may use your program to find such an example. (2 marks)
5. For 10 different values of n selected by you, use your program to compute $A(n)$, defined as follows. $A(n)$ is the average length of a longest common subsequence of three randomly generated binary sequences X , Y , Z of length n , where all symbols in the sequences are chosen independently and uniformly at random from $\{0, 1\}$ and the average is taken over 10 independent runs. In other words, you need to run your program 10 times for each n and therefore 100 times in total.

Present your results in a table with ten rows and three columns: one column for n , one column for $A(n)$, and one column for the ratio $A(n)/n$. According to your experiments, does $A(n)/n$ seem converge to any constant, and if so, what is its value? (3 marks)
6. This question concerns the case of two input sequences. Prove that every algorithm that outputs *all* longest common subsequences of two input sequences has a worst-case running time that is exponential. To do so, show how to define, for every positive integer n , two length- n sequences X_n , Y_n having $\Omega(c^n)$ different longest common subsequences, where c is a constant. You are allowed to use an alphabet of size n , i.e., the symbols in X_n , Y_n can come from a set $\{a_1, a_2, \dots, a_n\}$. (3 marks)