Computer Science C73 October 6, 2021
Scarborough Campus University of Toronto

Homework Assignment #5
(worth 4% of the course grade)
Due: October 27, 2021, by 11:59 pm

• ***You must submit your assignment through the Crowdmark system.*** *You will receive by email an invitation through which you can submit your work in the form of separate PDF docucments with your answers to each question of the assignment. To work with a partner, you and your partner must form a group on Crowdmark. Crowdmark does not enforce a limit on the size of groups.* ***The course policy that limits the size of each group to at most two remains in effect:*** *submissions by groups of more than two persons will not be graded.*
• *It is your responsibility to ensure that the PDF files you submit are legible. To this end, I encourage you to learn and use the LaTex typesetting system, which is designed to produce high-quality documents that contain mathematical notation. You are not required to produce the PDF files you submit using LaTex; you may produce it any way you wish, as long as the resulting document is legible.*
• *By virtue of submitting this assignment you (and your partner, if you have one) acknowledge that you are aware of the policy on homework collaboration for this course.*[a]
• *For any question, you may use data structures and algorithms previously described in class, or in prerequisites of this course, without describing them. You may also use any result that we covered in class, or is in the assigned sections of the official course textbooks, by referring to it.*
• *Unless we explicitly state otherwise, you may describe algorithms in high-level pseudocode or point-form English, whichever leads to a clearer and simpler description.*
• *Unless we explicitly state otherwise, you should justify your answers. Your paper will be graded based on the correctness and efficiency of your answers, and the clarity, precision, and conciseness of your presentation.*

[a] "In each homework assignment you may collaborate with at most one other student who is currently taking CSCC73. If you collaborate with another student on an assignment, you and your partner must submit only one copy of your solution, with both of your names. The solution will be graded in the usual way and both partners will receive the same mark. Collaboration involving more than two students is not allowed. **For help with your homework you may consult only the instructor, TAs, your homework partner (if you have one), your textbook, and your class notes. You may not consult any other source.**"

Recall that a dynamic programming algorithm to solve a given problem P involves the following elements:
(a) *A definition of a polynomial number of subproblems that will be solved (and from whose solution we will compute the solution to P — see (c) below).*
(b) *A recursive formula to compute the solution to each subproblem from the solutions to smaller subproblems. This induces a partial order on the subproblems defined in (a).*
(c) *A way to compute the solution to P from the solutions to the subproblems computed in (b).*

Proving the correctness of a dynamic programming algorithm amounts to justifying (i) why the recursive formula in step (b) correctly computes the subproblems defined in step (a), and (ii) why the computation in (c) yields a solution to the given problem. Part (ii) is often immediate from the definition of the subproblems.

Question 1. (10 marks) Let x be a string. A ***division*** of x is a sequence of non-empty strings $x_1, x_2, \ldots, x_k$ such that $x = x_1 \circ x_2 \circ \cdots \circ x_k$, where $\circ$ denotes concatenation. (Note that if x is the empty string then it has a single division, namely the empty sequence — i.e., $k = 0$.) A ***legal division*** of x is a division of x such that every string in the division is a word in the English language. (The sequence of words does not have to make sense for it to be a legal division; we only require that the individual words are legitimate English words.) For example, there are exactly four legal divisions of the string "`bananagram`": `banana-gram`, `ban-an-a-gram`, `ban-anagram`, `ban-a-nag-ram`; and there are no legal divisions of the string "`supercalifragilisticexpialidocious`".

Give a polynomial-time dynamic programming algorithm that takes as input a string $x[1..n]$ and returns the number of different legal divisions of that string. For example, if the input is "`bananagram`", the algorithm should return 4; if the input is "`supercalifragilisticexpialidocious`", the algorithm should return 0.

Justify the correctness of your algorithm and analyze its running time.

You may assume that you have access to a procedure $\textsc{Dict}(x, i, j)$ that, given a string $x[1..n]$ and indices i, j such that $1 \le i \le j \le n$, returns 1 if $x[i..j]$ is an English word and returns 0 otherwise. You may further assume that a call to this procedure takes $O(1)$ time.

Question 2. (10 marks) Let A and B be sequences. A is a ***supersequence*** of B if B is a subsequence of A. Recall that a subsequence of A is a sequence of elements of A that appear in the same order as in A; more precisely if $A = a_1, a_2, \ldots, a_n$, a subsequence of A is $a_{i_1}, a_{i_2}, \ldots, a_{i_k}$, where $1 \le i_1 < i_2 < \cdots < i_k \le n$. A ***palindrome*** is a (finite) sequence that is equal to its reversal — i.e., it reads the same backwards as forwards — for example, the inscription on the fountain in the courtyard of Agia Sophia: $\nu\iota\psi o\nu\alpha\nu o\mu\eta\mu\alpha\tau\alpha\mu\eta\mu o\nu\alpha\nu o\psi\iota\nu$ which (with inter-word spaces omitted) means "wash the sins, not just the face".

a. Give an $O(n^2)$ dynamic programming algorithm that takes as input a sequence $A[1..n]$ and returns the length of a shortest supersequence of A that is a palindrome. For example, if $A =$ `r,e,s,t,a,t,e` the algorithm should return 9 since the shortest supersequence of A that is a palindrome is `r,e,s,t,a,t,s,e,r`.

Justify the correctness of your algorithm.

b. Modify your algorithm in part (a) so that it returns a minimum-length supersequence of A that is a palindrome (instead of its length, as in part (a)).