Assignment 3

Due by Friday, June 15

Acknowledgments. Acknowledge all the sources you used to complete the assignment. DO NOT COPY! Please read `http://www.student.cs.uwaterloo.ca/~cs341/` for general instructions and policies.

For all algorithm design questions, you must give the algorithm, argue the correctness, and analyze time complexity.

1 Longest common subsequence of many strings [10 marks]

Recall that a string $z = z_1 z_2 \cdots z_k$ is a *subsequence* of a string $x = x_1 x_2 \cdots x_n$ if there are indices $1 \leq i_1 < i_2 < \cdots i_k \leq n$ for which $z_1 = x_{i_1}$, $z_2 = x_{i_2}$, …, $z_k = x_{i_k}$. The string z is a *common subsequence* of a set of m strings $x^{(1)}, x^{(2)}, \ldots, x^{(m)}$ if and only if z is a subsequence of all m of these strings. For example, the string `ONI` is a common subsequence of the 3 strings

$$\begin{aligned} x^{(1)} &= \texttt{POLYNOMIAL} \\ x^{(2)} &= \texttt{EXPONENTIAL} \\ x^{(3)} &= \texttt{CONSTANTTIME} \end{aligned}$$

but the string `PAL` is not.

Design an dynamic programming algorithm that, given as input a set of m strings $x^{(1)}, x^{(2)}, \ldots, x^{(m)}$ of lengths $n_1, \ldots, n_m$, respectively, outputs the length of the longest common subsequence of $x^{(1)}, x^{(2)}, \ldots, x^{(m)}$.

2 Bookshelf [10 marks]

In the LIBRARYBOOKSHELF problem, you are given as input a set of n books and the thickness $t_i > 0$ of each book $i \in \{1, 2, \ldots, n\}$, along with the width W of the bookshelves that will be used to build the library.

Design a greedy algorithm that determines the minimum number of bookshelves required to store the books, keeping in mind that since the books have call numbers, you must place them in order in the bookshelves, but that you can choose how many go on a shelf.

3 Hiring [10 marks]

In the Hiring problem, you own two companies that have salary budgets $B_1 > 0$ and $B_2 > 0$, respectively. There are n people $1, 2, \dots, n$ applying to work at one of your companies, where person i requires salary s_i and would generate revenues $r_1(i) > 0$ and $r_2(i) > 0$ to companies 1 and 2, respectively.

A valid solution to the Hiring problem is the maximum value R for which there exists a pair of disjoint sets $S_1, S_2 \subseteq \{1, 2, \dots, n\}$ that satisfy

(i) $\sum_{i \in S_1} s_i \le B_1$,

(ii) $\sum_{i \in S_2} s_i \le B_2$, and

(iii) $\sum_{i \in S_1} r_1(i) + \sum_{i \in S_2} r_2(i) = R$.

Design a dynamic programming algorithm to solve the Hiring problem. (Note that to solve the problem, the algorithm needs to find the maximum revenue R as described above, but it does not need to identify the sets S_1 and S_2.) In your description of the algorithm, make sure you clearly indicate what are the subproblems and the order in which they are solved. Also: your time complexity analysis should be in terms of n, B_1, and B_2.

4 Scriptio continua [10 marks]

Let $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \dots, \mathtt{z}\}$ denote the alphabet, and let $S \subset \Sigma^*$ be the set of all words in the English language. In the SriptioContinua problem, you are given a string $s \in \Sigma^n$ (that does not have any spaces or punctuation marks) and you must output `True` if there exists a sequence of words $w_1, \dots, w_m \in S$ such that $s = w_1 w_2 \cdots w_m$ is the concatenation of the words $w_1, \dots, w_m$, and `False` otherwise.

For example, for the input

$$s = \mathtt{thisstringisasequenceofwordswrittenwithoutanyspaces}$$

the valid solution is `True` since we can partition s into the sequence of 11 words

$$s = \underbrace{\mathtt{this}}_{w_1}\ \underbrace{\mathtt{string}}_{w_2}\ \underbrace{\mathtt{is}}_{w_3}\ \underbrace{\mathtt{a}}_{w_4}\ \underbrace{\mathtt{sequence}}_{w_5}\ \underbrace{\mathtt{of}}_{w_6}\ \underbrace{\mathtt{words}}_{w_7}\ \underbrace{\mathtt{written}}_{w_8}\ \underbrace{\mathtt{without}}_{w_9}\ \underbrace{\mathtt{any}}_{w_{10}}\ \underbrace{\mathtt{spaces}}_{w_{11}}$$

with $w_1, w_2, \dots, w_{11} \in S$.

1. Design a dynamic programming algorithm for solving the SriptioContinua problem. In your solution, you can assume that you have access to an algorithm IsWord that, on any input $w \in \Sigma^\ell$, outputs `True` if $w \in S$ and `False` otherwise, and has time complexity $\Theta(1)$. You should aim for an algorithm with runtime $O(n^2)$.

2. Modify the algorithm in part (a) so that when the answer is `True`, the algorithm also outputs $w_1, \dots, w_m \in S^m$ for which $s = w_1 w_2 \cdots w_m$.

5 Shipping paper [10 marks]

Note: For this assignment only, you do not need to program your solution to the problem; only the written solution needs to be submitted.

There are two warehouses, V and W, which ship supplies of paper to n factories, denoted $F_1, \ldots, F_n$. The factory F_i requires exactly d_i units of paper, it costs v_i to ship a unit of paper from V to F_i, and it costs w_i to ship a unit of paper from W to F_i, for each $i \in \{1, 2, \ldots, n\}$. The total amount of paper available at V and W is r_V and r_W, respectively, where $r_V + r_W = d_1 + \cdots + d_n$.

We want to determine how much paper to ship from V and W to each of the factories so that the total shipping cost is minimized. Formally, let x_i denote the number of units of paper that are shipped from V to F_i and y_i denote the units of paper that are shipped from W to F_i, for $1 \leq i \leq n$. The following constraints must be satisfied:

- $x_1, \ldots, x_n, y_1, \ldots, y_n$ are non-negative integers,
- $x_1 + \cdots + x_n = r_V$,
- $y_1 + \cdots + y_n = r_W$, and
- $x_i + y_i = d_i$ for each $i \in \{1, 2, \ldots, n\}$.

The total shipping cost is

$$C = \sum_{i=1}^{n} (v_i x_i + w_i y_i).$$

An instance of the Shipping problem is specified by the values r_V, r_W, $d_1, \ldots, d_n$, $v_1, \ldots, v_n$, and $w_1, \ldots, w_n$. A valid solution to such an instance is an n-tuple $(x_1, \ldots, x_n)$ that satisfies the constraints above and minimizes C. (Since the value of x_i determines the value of y_i uniquely for each $i \in \{1, 2, \ldots, n\}$, the n-tuple $(y_1, \ldots, y_n)$ does not need to be provided explicitly in the solution.)

Design a greedy algorithm that solves the Shipping problem.