ASSIGNMENT 5

1. The final essay of a course is due. As it is often the case, there is a minumum number of pages to be handed in, with text formatted at 72-80 columns. That is, a line break must occur between columns 72 and 80, inclusive, unless the there is no blank space in which to break the line, in which case lines shorter than 72 columns are allowed.

    For example, consider a small example in which text must be formatted to between 10-15 columns:

    ```
    123456789012345

    The quick red                 ..1
    fox is                        ..2
    exhausted                     ..3
    ```

    Observe that the line break in line (2) is at a length shorter than 10 characters since placing the word exhausted in line (2) would exceed 15 columns.

    A student wishing to reduce the amount of writing needed meet the number of pages. She is writing a program that **maximizes** the number of lines used to format a given text, that is to maximize the number of pages when the final essay is printed.

    Give an algorithm for this and prove its correctness. [10 points]

2. This problem is about choosing line breaks to format a paragraph. Tim Fowl is a student who believes that the more lines that end with the letter R the higher his marks will be. Suppose that the paragraph has words of lengths $l_1, l_2, \ldots, l_n$ in that order. Each pair of consecutive words must be separated by a blank, and a blank has length 1. Just as in the previous question line breaks must appear between column 72 and 80 unless including the next word would exceed 80 columns.

    Devise an algorithm that maximizes the number of lines finishing in the letter R, while formatting acording to the 72-80 column specification.

    For example, here are two different formatting choices:

    ```
    I propose to consider the
    question, do you want to super size that,
    sir. This should begin with definitions of
    the meaning
    of the terms super and size.

    I propose to consider
    the question, do you want to super
    size that, sir.
    This should begin with definitions of
    the meaning of the terms super
    and size.
    ```

    Give an efficient algorithm to format the text in a way that maximizes the number of lines finishing with the letter R. [10 points]

3. You are given two strings $X = x_1 \ldots x_n$ and $Y = y_1 \ldots y_m$, over the alphabet Σ. Each character has an associated gain $G(c)$. Give an algorithm that finds the common subsequence of X and Y which maximizes the overall gain, i.e. the sum of the gains of the individual characters in the common subsequence chosen. [10 points]

4. In class we saw an algorithm for finding the shortest path between two vertices u and v in a weighted graph. Consider a delivery person which gets paid by the mile and hence wishes to find the most circuitous route to destination that does not repeat vertices or edges. The decision version of this problem is "does there exist a path between u and v whose length is at least L". Show that the decision version of this problem is NP-complete. [10 points]

5. This question is inspired by UW policies on exam scheduling, in particular, the rule that the university will "strive to schedule final exams with no student having two exams in a row...," among other constraints. To appreciate the difficulty of this task, we will look at one (simplified) abstraction of the problem. Specifically, we only have one classroom (!), and we want to schedule n classes $c_1 \dots c_n$ into n different time slots $1 \dots n$. The schedule can be specified by a one-to-one mapping $f : \{c_1 \dots c_n\} \to \{1 \dots n\}$. We are given a list of ℓ students, and for the j-th student, we are given a list S_j of the classes that he/she takes. We want to ensure that no two classes in S_j are assigned to two consecutive time slots. The precise statement of this problem is thus the following:

> *Input*: a set of elements $c_1 \dots c_n$, and a collection of subsets $S_1 \dots S_\ell \subseteq \{c_1 \dots c_n\}$.
> *Output*: "yes" iff there exists a one-to-one mapping $f : \{c_1 \dots c_n\} \to \{1, \dots, n\}$ such that for every j and for every $x, y \in S_j$, we have $f(y) \neq f(x) + 1$ and $f(y) \neq f(x) - 1$.

Prove that this problem is NP-complete. [10 points]

Hint: Reduce from the HAMILTONIAN-PATH problem (a variant of the HAMILTONIAN-CYCLE problem where we are to decide the existence of a path that visits all vertices exactly once in a given undirected graph). You may assume that HAMILTONIAN-PATH is NP-complete.

6. Consider the interval scheduling problem we saw in class, where we wish to schedule as many disjoint activities as possible from a set of overlapping activities. An example of this would be a movie critic during the *Toronto International Film Festival* (TIFF) wishing to attend as many movies as possible with the proviso that there can be no two overlapping shows, i.e. the movie critic can attend at most one movie at any given time.

Suppose now that the movie critic has personal preferences for each movie and now wishes to maximize the sum of the preferences over the movies watched.

More formally, we have a set of activities $A = \{a_1, a_2, \dots, a_n\}$, where each activity a_i has an associated start time s_i, a finishing time f_i and a preference p_i. The goal is to find a subset S of nonoverlapping activities such that

$$\sum_{i \in S} p_i$$

is maximized.

   (a) Show with an example that the greedy method does not work. [5 points]
   (b) Give a dynamic programming algorithm for this problem. [10 points]

7. Consider now a movie critic which is attending the *Movies with Sequels Film Festival*, which showcases movies and their sequels. The showings could be, for example, Superman I..IV, Star Wars I..VI, Spider-Man 1-3, Star Trek I-VI, etc.

Movies in a single series are shown with variable size breaks in between. For example Spider-Man 1 is on Tuesday from 9-11am, Spider-Man 2 also on Tuesday from 1-3pm and Spider-Man 3 on Wednesday from 9:30-11:30am.

The critic in this case has no preferences as all sequels are equally bad, but s/he still needs to watch as many movies as possible, with the proviso that it either watches an entire series of none of the movies in it.

Show that in this case the decision version of the interval scheduling problem is NP-complete. Please state the decision version first explicitly, then show that it is NP-complete. [12 points]