Date: 18.10.2024

(CS1.301) Algorithm Analysis and Design (Monsoon 2024)

Quiz 2

Total marks: 25

Alloted time: 60 minutes

Instructions:

- There are a total of 2 questions with varying credit.
- Discussions amongst the students are not allowed. No electronic devices with internet may be used. You however may use your notes to work things out.
- Any dishonesty shall be penalized heavily.
- Place your identity cards on the table for verification.
- Be clear in your arguments. Vague arguments shall not be given any credit. All your answers **must** use Dynamic Programming paradigm.

Question 1 [10 marks]

A currency has currency notes in the following denomination: $1, 4, 7, 13, 28, 52, 91, 365$. Your goal is to design an algorithm for finding the smallest number of notes that can be used for a given amount of currency. For example, for an amount of 9, we can either give out 3 notes with denominations either $1, 4, 4$ or $1, 1, 7$, but no solution exists with 2 notes.

Design and analyze a dynamic programming algorithm for finding the smallest number of notes.

Assume that you will be asked to give out notes only for an amount smaller than S. Assume sufficient supply of notes for each denomination, and a denomination can be used multiple times.

Question 2 **[15 marks]**

Suppose we want to *typeset* a paragraph of text onto a piece of paper (or if you insist, a computer screen). The text consists of a sequence of n words, where the i^{th} word has length $\ell[i]$. We want to break the paragraph into several lines of total length exactly L.

Depending on how the paragraph is broken into lines of text, we must insert different amounts of white space between the words.

The paragraph should be fully justified, meaning that

- **the first character on each line starts at the left margin**, and
- except for the last line, **the last character on each line ends at the right margin.**

There must be at least one unit of white space between any two words on the same line. We need not add space after the last word in the line.

Define the **slop** of a paragraph layout as the sum over all lines, except the last, of the square of the amount of extra white-space in each line, not counting the one unit of required space between each adjacent pair of words.

Specifically, if a line contains words i through j, then we need that

$$(j-i)+\sum_{k=i}^{j} \ell[k] \leqslant L \tag{1}$$

where $(j-i)$ is contributed by the mandatory space between the words in the line and, the **slop** of that line is defined to be

$$\left(L-(j-i)-\sum_{k=i}^{j} \ell[k]\right)^2 .$$

Design an and analyze an efficient **dynamic programming** algorithm to break a list of words into lines while minimizing the total slop, i.e., the sum of slop on each line except the last. (The last line must still satisfy the constraint Eq. (1).)

Your algorithm will be given a list of word lengths and will output the minimum slop achieved. Note that we are not interested in the number of lines, and we have to put words into only one paragraph. Further, the order of the words cannot be changed.

Hint: *Consider the situation where optimal arrangement of first i words for all $i < j$ gives you a way to find an optimal arrangement for the first j words.*