CS 4349.400 Homework 4

Due Wednesday September 26th, in class

Remember the following:

Greedy algorithms never work!
and
Dynamic programming is not about filling tables. It's about smart recursion!

Now, please answer each of the following questions.

1. Suppose we want to display a paragraph of text on a computer screen. The text consists of a sequence of n words, where the ith 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 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.

   Define the *slop* of a paragraph layout as the sum over all lines, *except the last*, of the cube 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 the slop of that line is defined to be $(L - j + i - \sum_{k=i}^{j} \ell[k])^3$. Describe a dynamic programming algorithm to compute the minimum slop possible for the given words. Give asymptotic bounds for both the space and time required by your algorithm. ***Simply giving an iterative algorithm without first describing the recursive procedure or function leading to it is worth* zero *credit.***

2. You and your eight-year-old nephew Elmo decide to play a simple card game. At the beginning of the game, the cards are dealt face up in a long row. Each card is worth a different number of points. After all the cards are dealt, you and Elmo take turns removing either the leftmost or rightmost card from the row, until all the cards are gone. At each turn, you can decide which of the two cards to take. The winner of the game is the player that has collected the most points when the game ends.

   Having never taken an algorithms class, Elmo follows the obvious greedy strategy—when it's his turn, Elmo *always* takes the card with the higher point value. Your task is to find a strategy that will beat Elmo whenever possible. (It might seem mean to beat up on a little kid like this, but Elmo absolutely *hates* it when grown-ups let him win.)

   (a) Prove that you should not also use the greedy strategy. That is, show that there is a game that you can win, but only if you do *not* follow the same greedy strategy as Elmo.

(b) Describe and analyze an algorithm to determine, given the initial sequence of cards, the maximum number of points that you can collect playing against Elmo. You may assume that you get the first turn.

(c) Five years later, thirteen-year-old Elmo has become a *much* stronger player. Describe and analyze an algorithm to determine, given the initial sequence of cards, the maximum number of points you can collect playing against a *perfect* opponent. You may assume that you get the first turn.

For parts (b) and (c), give asymptotic bounds for both the space and time required by your algorithm.

3. You are driving a bus along a highway, full of rowdy, hyper, thirsty students and a soda fountain machine. Each minute that a student is on your bus, that student drinks one ounce of soda. Your goal is to drop the students off quickly, so that the total amount of soda consumed by all students is as small as possible.

You know how many students will get off of the bus at each exit. Your bus begins somewhere along the highway (probably not at either end) and moves at a constant speed of 37.4 miles per hour. You must drive the bus along the highway; however, you may drive forward to one exit then backward to an exit in the opposite direction, switching as often as you like. (You can stop the bus, drop off students, and turn around instantaneously.)

Describe an efficient algorithm to drop the students off so that they drink as little soda as possible. (An algorithm that merely computes the minimum amount of soda consumed is worth full credit.) Your input consists of the bus route (a list of exits, together with the travel time between successive exits), the number of students you will drop off at each exit, and the current location of your bus (which you may assume is an exit). Give asymptotic bounds for both the space and time required by your algorithm.