CPSC 320: Intermediate Algorithm Design and Analysis
Assignment #4, due Monday, June 8^{th}, 2015 at 2:15pm in Room x235, Box 2

[15] 1. After the end of term 2[1], a students decides to ride her bicycle from Vancouver to Halifax. She starts on the road at kilometer 0. Along the way there are n hotels $H_1, \ldots, H_n$, at distances $h_1 < h_2 < h_3 < \ldots < h_{n-1} < h_n$ from the trip's starting point respectively. The only places where the student is willing to stop for the night are these hotels, but she can choose which hotels she stops at. She must stop at the final hotel H_n since this is her destination (she is planning on flying back).

Ideally, this student would like to travel 100Km per day, but this may not be possible, depending on the spacing of the hotels. In order to determine how close the student got to this target distance, she defines the *penalty* for a day to be $|100 - x|^{1.5}$ where x is the distance traveled that day. Her objective is to plan her trip in a way that minimizes the sum of the daily penalties over all days of the trip.

- [3] a. One possible solution would be to use a greedy algorithm that determines where to stop on day i by choosing the hotel whose coordinate is closest to $x_{i-1} + 100$, where x_{i-1} is the coordinate of the hotel that the student stopped at on day $i - 1$.

  Give an example to show that this greedy algorithm does **not** always succeeds in minimizing the sum of the daily penalties over all days of the trip.
- [4] b. Let D_k be the minimum sum of the daily penalties, over all possible trips that end at hotel H_k (that is, the student does not go any further). Write a recurrence relation for D_k, and explain why your recurrence relation is correct.
- [6] c. Using this recurrence relation, describe a dynamic programming algorithm that computes D_n. Your algorithm should return the list of hotels at which the student will spend the nights to achieve the minimum sum of daily penalties (not only the value D_n).
- [2] d. Analyze the running time of your algorithm as a function of n.

[1] There isn't enough time at the end of term 1. Besides, cycling across Canada at the end of term 1 would be a bad idea.

[15] 2. It is nearing the end of the semester and you are taking n courses, each with a final project that still has to be done. Each project will be graded on a scale from 1 to 100. Your goal, of course, is to maximize your average grade (which is the same as maximizing the sum of the grades) on the n projects.

You have a total of $H > n$ hours in which to work on the n projects (all of which are due at exactly the same time) and you want to decide now to divide up this time. For simplicity, assume that H is a positive integer, and that you will spend an integer number of hours on each project. You have come up with a set of functions $f_1, \ldots, f_n$ (a rough estimate, of course) such that working x hours on project i will get you a grade of $f_i(x)$ on that project – you may assume that $f_i(x)$ does not decrease if x increases.

So, given H and $f_1, \ldots, f_n$, you need to determine how many (integer) hours you will spend on each project so your average grade is as large as possible.

[5] a. Let $G[h, j]$ be the sum of the grades you will obtain by spending h hours in total on the projects for courses $1, 2, \ldots j$. Write a recurrence relation for $G[h, j]$, and explain why your recurrence relation is correct.

[8] b. Using this recurrence relation, describe a dynamic programming algorithm that computes $G[H, n]$. Your algorithm should return the amount of time you will spend on each course project.

[2] c. Analyze the running time of your algorithm as a function of H and n.

[1] 3. (Bonus) How long did it take you to complete this assignment (not including any time you spent revising your notes before starting)?