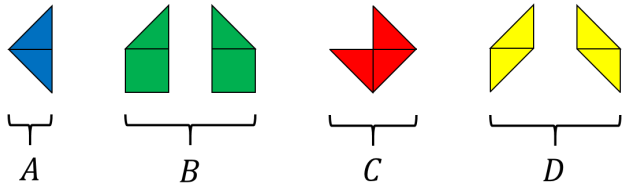
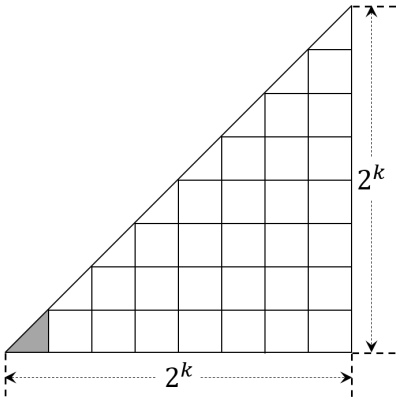
Homework #1

(Due: Mar 5)



(a) Four types of tiles



(b) A right triangular grid to tile ($k = 3$)

Figure 1: Tiling a right triangular grid.

Task 1. [70 Points] Tiling a Triangular Grid

Given an isosceles right triangular grid for some $k \geq 2$ as shown in Figure 1(b), this problem asks you to completely cover it using the tiles given in Figure 1(a). The bottom-left corner of the grid must not be covered. No two tiles can overlap and all tiles must remain completely inside the given triangular grid. You must use all four types of tiles shown in Figure 1(a), and no tile type can be used to cover more than 40% of the total grid area. You are allowed to rotate the tiles, as needed, before putting them on the grid.

(a) [**25 Points**] Design and explain a recursive divide-and-conquer algorithm for tiling the grid under the constraints given above. Include pseudocode.

(b) [**25 Points**] Write down recurrences describing the running time of your algorithm from part (a), and solve them.

(c) [**20 Points**] Write down recurrences for counting the number of tiles of each type used by your algorithm, and solve them to show that no tile type covers more than 40% of the total grid area.

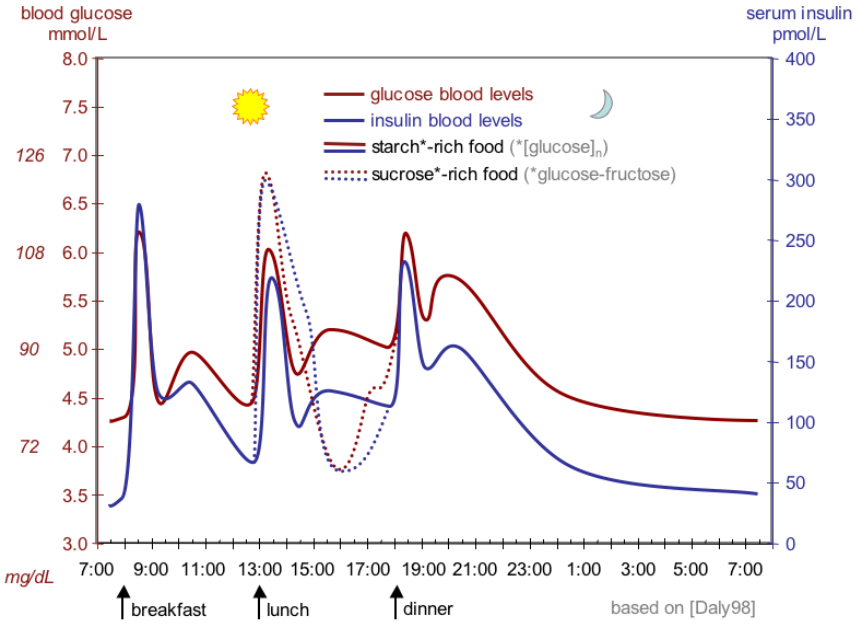


Figure 2: Idealized curves of human blood glucose and insulin concentrations during the course of a day containing three meals; in addition, effect of sugar-rich meal is highlighted. [*Source: Jakob Suckale, Michele Solimena – Solimena Lab and Review Suckale Solimena 2008 Frontiers in Bioscience PMID 18508724, preprint PDF from Nature Precedings, original data: Daly et al. 1998 PMID 9625092.*]

Task 2. [50 Points] Blood Sugar

Carbohydrates are abundant in many food items including whole grains, fruits, vegetables, breads and rice, and they are one of the most important sources of eneregy for our body. Our digestive system breaks down carbohydrates into glucose which is then converted into energy to support our bodily functions.

When we eat a food containing carbohydrates, our blood glucose level rises. This rise in blood sugar causes the pancreas to produce a hormone called insulin that triggers absorption of blood sugar by cells for producing energy or for storage. With the absorption of blood sugar, concentration of sugar in our bloodstream drops. As the blood sugar level drops, the pancreas starts making another hormone called glucagen that prompts the liver to release its stored sugar. This interplay between insulin and glucagen keeps the concentration of sugar in our blood within safe limits. Figure 2 shows idealized curves of human blood glucose and insulin concentrations during the course of a day. When our body cannot make insulin or cannot use the insulin it produces properly, type 2 diabetes develops.

In this task, for a given food item $\mathcal{F}$ and a given healthy person $\mathcal{P}$, you will be given the following pieces of information.

– An idealized sequence $\langle s_0, s_1, s_2, \ldots, s_{n-1} \rangle$ of elevations in blood sugar level (in mg/dL) of $\mathcal{P}$ after the consumption of 1 gm of $\mathcal{F}$, where for $0 \leq i < n$, s_i is the average elevation in blood sugar level during hour i after consuming $\mathcal{F}$.

– For a period Π of n hours, the amount f_i of $\mathcal{F}$ consumed (in gm) by $\mathcal{P}$ during hour i of that period, where integer $i \in [0, n)$.

Your task will be to predict the total elevation t_i in blood sugar level of $\mathcal{P}$ in each hour $i \in [0, n)$ of Π resulting from the consumption of $\mathcal{F}$ as described above. For example, if $s_0 = 10$, $s_1 = 7$, $s_2 = 3$, $f_0 = 2$, $f_1 = 1$ and $f_2 = 3$ then $t_0 = 20$, $t_1 = 24$ and $t_2 = 43$.

Give an efficient algorithm for computing all t_i's for the entire period Π, and derive an asymptotic upper bound on its running time. Your algorithm must run in $\mathcal{O}(n \log n)$ time.

Task 3. [20 Points] Multiplying Long Integers of Significantly Unequal Lengths

Recall that for $n \geq m$ multiplying an n-bit number with an m-bit number using the standard grade-school algorithm takes $\Theta(mn)$ time while standard Karatsuba's algorithm takes $\Theta\left(n^{\log_2 3}\right)$ time. For $m = o\left(n^{\log_2 3 - 1}\right)$, the grade-school algorithm runs asymptotically faster than Karatsuba's. This task asks you to fix this problem.

(a) [**5 Points**] Argue that when $n \geq 2m$, Karatsuba's algorithm can get away with using only two (2) recursive multiplications of smaller numbers instead of three (3).

(b) [**15 Points**] Modify Karatsuba's recursive divide-and-conquer algorithm to take advantage of the observation from part (a) at each level of recursion (if applicable). Write down the recurrene relation(s) describing the running time of the modified algorithm, and solve them. What bound do you get? What is the smallest asymptotic value of m below which the modified algorithm loses to the grade-school algorithm again?

Task 4. [40 Points] Transporting Prioritized Weighted Boxes

You have n weighted prioritized boxes with w_i being the weight of the i-th box and p_i being its priority, where $i \in [1, n]$, and both w_i and p_i are real numbers. No two boxes have the same priority. You have a truck that cannot carry more than W weight, and you want to load it with as many boxes as possible in order of priority (highest to lowest) without exceeding its weight limit. You cannot load a box unless all boxes with higher priority have already been loaded. For example, if $w_1 = 2$, $p_1 = 9.5$, $w_2 = 5$, $p_2 = 15$, $w_3 = 1$, $p_3 = 5$, $w_4 = 4$, $p_4 = 12$ and $W = 10$, then you can only load box 2 and box 4.

(a) [**25 Points**] Assuming that the boxes are not initially sorted based on priority, design a recursive divide-and-conquer algorithm that can efficiently (see part (b)) identify all boxes you must load into your truck. Include pseudocode. Prove that your algorithm is correct.

(b) [**15 Points**] Write down the recurrence relation desribing the running time of your algorithm from part (a), and solve it to show that your algorithm runs in $\Theta(n)$ time.