Algorithms and Data Structures
MSIS 26:198:685
Homework 2

Instructor: Farid Alizadeh
Due Date: Monday October 23, 2019, at 11:50PM

last updated on October 12, 2019

Please answer the following questions in **electronic** form and upload and submit your files to Sakai site, before the due date. Make sure to push the **submit** button.

You can typeset your answer using MS Word (and MS equation for mathematical formulas), or LaTeX, or you may simply handwrite your note and scan and turn it not a **single pdf format file** and upload to Sakai.

1. Solve the following Recurrence relations. Show your work. Express the solution in terms of big Θ of familiar functions.

   1a) $F(n) = 3F(n/3) + n^2$, with $F(1) = 1$

   1b) $G(n) = 2G(\sqrt{n}) + \log_2(n)$, with $G(1) = 1$

   1c) $H(n) = \frac{H(n-1)+H(n-2)+\cdots+H(1)}{n} + 2$, with $H(1) = 1$

   1d) $f(n) = 4f(n/2) + \frac{n^2}{\log(n)}$.

2. Recall that in boolean algebra we only have the two values of 1 (also known as `True`) and 0 (also known a `False`.) The two main operations on boolean values are product '$\cdot$' (or `And`) and sum '+' (or `Or`) defined as follows:

$$a \cdot b = 1 \text{ if both } a = 1 \text{ and } b = 1, \text{ otherwise } a \cdot b = 0$$
$$a + b = 1 \text{ if at least one of } a \text{ or } b \text{ is 1, otherwise } a + b = 0.$$

   With these definitions we can define the *boolean matrix product* $AB = C$ where
$$C_{ij} = A_{i1} \cdot B_{1j} + \cdots + A_{in} \cdot B_{nj}.$$

   Consider two $n \times n$ matrices A and B with boolean, that is only 0 or 1 entries. Define the *boolean product* of these two matrices the same way that ordinary numerical matrices are multiplied, except that instead of multiply we use the above boolean product, and instead of plus, we use the above boolean sum.

   2a) To compute the boolean matrix product AB, explain why the Strassen's algorithm cannot be used directly on the boolean matrices using only the boolean operations.

   2b) Interpret 0's and 1's as *integers*, and use ordinary multiplication. How can you extract boolean multiplication from the result of ordinary integer multiplication? Strassen's algorithm for this case?

   2c) Arithmetic complexity is not appropriate here because the original input is a matrix of bits. Give a complete *bit* complexity analysis. Assume in part a) you used a Strassen-like algorithm with arithmetic complexity $\mathcal{O}(n^{2+\omega})$ where $0 < \omega < 1$.

3. A list of numbers is *unimodal* if it starts with a sequence of *non-decreasing* numbers followed by a sequence of *non-increasing numbers*. Each subsequence can be empty. For instance, the Following sequences are unimodal: (1, 2, 7, 6, 4), (1, 1, 2, 2, 5,4), (2,2,2,2), (4,3,2,1). However, the following sequences are *not* unimodal: (1,2,3,2,4), (1,1,2,2,1,2).

   Given an unimodal sequence, give the most efficient algorithm you can to find its maximum element, both its value, and its index in the list. If there are more than one occurrence of the maximum, return the smallest index in which it occurs. Provide a complete complexity analysis, deriving any difference equation needed. Solve the difference equation you derived to find the time complexity of your algorithm.

4. Apply Euclid's algorithm to:

   4a) find the $\gcd(726, 693)$. Also find the integers x, y such that $\gcd(726, 693) = 726x + 693y$.

   4b) Either determine that the following equation does not have a solution, or find x: $77x = 4 \mod 20$.