CS 180, Fall 2015 Homework 1

The following homework is due on Wednesday, Oct 7th by the beginning of lecture.

When submitting your homework, please include your name at the top of each page. If you submit multiple pages, please staple them together.

1. In class, we discussed the game of NIM. This game begins with a placement of n rows of matches on a table. Each row i has m_i matches. Players take turns selecting a row of matches and removing any or all of the matches in that row. Whoever claims the final match from the table wins the game. This game has a winning strategy based on writing the count for each row in binary and lining up the binary numbers (by place value) in columns. We note that a table is *favorable* if there is a column with an odd number of ones in it, and the table is *unfavorable* if all columns have an even number of ones.

   Example: Suppose we start off with three rows of matches of 7, 8 and 9.
   $7 = 0111$ (binary)
   $8 = 1000$ (binary)
   $9 = 1001$ (binary)
   Since the third row from the right and the second row from the right, each have odd numbers of ones, the table is favorable.

   (a) Prove that, for any favorable table, there exists a move that makes the table unfavorable. Prove also that, for any unfavorable table, any move makes the table favorable for one's opponent. Write the algorithm that on input of favorable table output the row and the number of matches to remove from the row, to make the table unfavorable.

   (b) Give a winning strategy for this game.

2. A polynomial of degree n is of the form $P(x) = a_n x^n + a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + ... + a_1 x + a_0$ where $a_0, a_1, ..., a_n$ are some constants called the *coefficients* and x is a variable.

   (a) Assume that you have two sub-procedures for addition and multiplication. Describe an algorithm that takes as input the coefficients $a_0, a_1, ..., a_n$ and value v, and computes the value of $P(v)$. What is the total number of multiplication and additions performed by your algorithm?

   (b) Can you design an algorithm that only requires $O(n)$ multiplications and additions? (hint: consider a recursive solution that reduces the degree of the polynomial.)

3. Given an algorithm that computes the integer division (i.e "floor") of two numbers a/b if $a < 10b$, and division by any constant, i.e. 10. Give an algorithm that divides any two integers. You do not have enough memory to keep a decimal representation of a and b. All you have is a constant number of variables that can store numbers. No "fancy" encoding. Give an efficient algorithm in a recursive and iterative version. (Hint: we want to see a $\log_{10} n$ complexity (no "bits" operations) algorithm).
   For instance, in class we saw a division algorithm that does not use representation: It subtracts b from a until the result is less than b and counts the number of time it did this subtraction. We want the same flavor algorithm. The algorithm is what is called "long division" in elementary school, but here we do not use representation.

4. There is a multi-day tennis tournament with $n = 2^k$ players. On a given day, each player plays one match, so $n/2$ matches are played per day. Over the course of the tournament, each player plays against every other player exactly once, so a total of $n(n-1)/2$ matches will be played.

   Write a recursive algorithm to schedule these matches in the minimum number of days. It should take as input a number $n = 2^k$ and return as output a table with $n - 1$ rows, in which row i is a list of the $n/2$ matches to be played on day i.