Final | Purdom
5-4-99 | B501

1. Suppose you have a set of n items.
   a. How many subsets can be formed from the n items? Don't forget to include the empty set and the set of all items.
   b. Some of these subsets have an even number of items. Some have an odd number. How many subsets are there of each kind?
   c. For some values of n the number of even subsets is the same as the number of odd subsets. Thus, for $n = 1$ the empty set has an even number of items (zero) and the complete set has an odd number (one). For which n do you have the same number of even and odd subsets? For which n do you have different numbers? (Carefully consider each integer n for zero upwards.)

2. Consider a Strassen Matrix Multiplication Algorithm that multiplies an m by n matrix times an n by p matrix such that the running time is given by the following recurrence

$$T_{m,n,p} = 7T_{m/2,n/2,p/2} + 4mn + 4np + 7mp.$$

For small problems, this algorithm will call a classical Matrix Multiplication Algorithm that uses time

$$T_{m,n,p} = amnp + bmp.$$

(Lower order terms have been omitted to reduce the amount of calculation you need to do.) Assume that m, n, and p are all even.
   a. Write an inequality such that when the inequality is satisfied it is better for the Strassen algorithm to call the classical algorithm rather than to do an other recursive call. For this part only, there is no need to simplify your answer. You are encouraged to leave it in the first form that you obtain.
   b. Simplify the answer from the previous part as much as appropriate (read the rest of the question) and then explain how the simplified inequality can be used in a fast Strassen program for non-square matrices. (You do not need to discuss the effects of the approximations made in the analysis. I will assume that if you have unlimited time you would do a more complete analysis.)

3. Simplify $\sum_{1 \le i \le n} H_{2i}$, where H_j is the jth harmonic number.

4. Give a good approximation for: $\sum_{1 \le i \le n} i^{-3}$.

5. Solve the equation $x^a \ln x = t$, for x as a function of t. Assume that a is fixed, $a \ge 1$, and that t is large.

6. This question is concerned with how much time is needed by the Euclidean Algorithm (Algorithm 1.5) in the text when the input (m and n) are two consecutive Fibonacci numbers. Suppose the input is F_i and F_{i+1}. How many remainders are computed (as a function of i)?