Homework II, Advanced algorithms, Spring 1995

This homework is due Tuesday May 2 at the beginning of the lecture. I am aware that below is a rather difficult set of problem and thus grading will be good natured and solutions to only few problems give good credit towards passing the course.

1. Suppose we are given a binary linear congruential shift register, i.e. a device with holds n binary values $b_1, b_2, \ldots b_n$ and which is updated according to the following rule $b_{i+1}^{new} = b_i^{old}$, $i = 1, \ldots n-1$ while $b_1^{new} = b_n^{old} + \sum_{i=1}^{n-1} a_i b_i^{old}$ where a_i are some fixed constants (which are not all 0). Consider the following problem: Given initial values $b_i^{(0)}$ and final values $b_i^{(1)}$, how many updates were needed to go from one to the other? Design an efficient algorithm that solves this problem, by efficient in this case is an algorithm that runs in time significantly faster than 2^n.

   Hint: Try to relate the problem to a problem we studied in the course.

2. (2p) We are interested in finding a generator in Z_p (the integers mod p). First give a formula for the number of such generators and try find a lower bound on their number (this lower bound might be on a form similar to $p/2$, $p/(\log p)$ or p^α for some number α). Secondly, show how to quickly find a number that probably is a generator. In particular for any T give an algorithm that in expected time polynomial in T and $\log p$ finds a number such that the probability that this number is a generator is $1 - \frac{1}{T}$. The probability is taken over the random choices of the algorithm. You may use the fact that Z_p is cyclic, i.e. that there is at least one generator.

   Hint: If g is a generator then a number y is not a generator if it is on the form g^x where $GCD(x, p-1) > 1$.

3. Suppose you are given a representation for $GF[2^n]$, i.e. you are given an irreducible polynomial of degree n. Design and analyze an efficient algorithm for division in $GF[2^n]$ under this representation.

4. (2p) Suppose you are given two representations for $GF[2^n]$ i.e. you are given two irreducible polynomials $f_1(\theta)$ and $f_2(\varphi)$ of degree n. Your task is to efficiently construct an isomorphism between the two representations.

   Hint: Essentially all you need do is to find a root of $f_1(x) = 0$ in the representation given by f_2 and then map θ do this number and extend to give an isomorphism. There are a number of details to check.

5. Suppose that you are given k different rational numbers $x_i = p_i/q_i$ and a number Q. It is a well known theorem that there is a denominator $q < Q$ such that for some numbers b_i, $|x_i - \frac{b_i}{q}| \leq Q^{-(1/k)} q^{-1}$. Use Lovasz basis reduction algorithm to efficiently find a denominator that is almost as good (i.e. matches the above bound except for a factor depending only on k).

6. (3p) In the lecture when using the basis reduction algorithm for polynomial factorization, I did not have a basis for the lattice L, but only a set of generators (i.e. with had more than n vectors). There are two ways around this. 1. (1p) We can by examining the set of generators given more closely construct a basis instead. 2. (2p) We can extend the basis reduction algorithm to run also on an input that contains more vectors. Try both approaches.

   Hint: In the case 2, if we have $m > n$ vectors but forget this and set up the algorithm as usual some b_i^* will be 0. However, being 0 is a special case of being very small. Also, once a vector is all 0 it can just be dropped.