CSE 208A Homework 1
Randomized algorithms
Due Tuesday, May 7

For each of the algorithm problems, design as asymptotically efficient an algorithm as possible. Give a correctness argument (explanation, if it is relatively simple, or proof if not) and time analysis. You may use any well-known algorithm or data structure, or algorithm from the text or from class, as a sub-routine without needing to provide details.

Plentiful pigeons for finding quardratic non-residues: Exercise 14.8 of MR, page 405.

Plentiful pigeons for irreducible polynomials: A polynomial is *irreducible* if it cannot be factored into a product of smaller degree polynomials. Give (and analyze) a probabilistic algorithm that given input a prime number p, generates a uniformly chosen random irreducible degree 2 polynomial over Z_p, i.e., a polynomial ax^2+bx+c that cannot be written as $(ex+f)(gx+h)$ over Z_p. You will probably need to be able to test quadratic residousity; see page 404 of MR. You need to give a bound on the expected running time of your algorithm.

Achieving the Average for Max Bisection: Consider the Max Bisection problem: Given a weighted, undirected graph $G = (V, E)$ with an even number of nodes, find a partition of V into two sets S, T with $|S| = |T|$ that maximizes the total weights of edges in $Cut(S, T)$. Give a *deterministic* algorithm that is guaranteed to find solutions to this problem within a factor of 2 of optimal. (Hint: this is very close to the Max Cut problem. The algorithms might also be very close.)

Sampling for Set Union: Give a probabilistic algorithm that is $O(n)$ expected time and uses as little as possible external memory to solve (with high probability) the following problem: Given two sets of exactly n elements each, S_1 and S_2, and a number $1 > \epsilon > 1/\log n$, approximate the size of $|S_1 \cup S_2|$ to within a factor of $1 + \epsilon$, i.e., with probability say at least 90%, your algorithm should output a value s with $|S_1 \cup S_2| \leq s \leq (1 + \epsilon)|S_1 \cup S_2|$. (Note: S_1 and S_2 are given as lists of elements with no repetitions in each list, but elements in S_1 may be equal to elements in S_2. S_1 and S_2 are not given in sorted order. Assume elements of S_1, S_2 are represented as k bit strings, and that comparisons, arithmetic operations and inner products mod 2 on k bit inputs are all constant time. In other words, k is the "word size" for the machine model. Elements of S_1, S_2 are representable as single words, and operations on single words are constant time. It follows that $k \geq logn$.)