CS265/CME309, Fall 2019. Instructor: Gregory Valiant

Problem Set 2

Electronic submission via Gradescope (submission code 92EK55) due **11:59pm Tuesday 10/8**. You are strongly encouraged to submit a homework with a partner—that is, submit one homework with both of your names.

[You may discuss these problems with classmates. Feel free to look at wikipedia, course notes, etc. for reference material, but do not try to specifically search online for solutions to the problems. ***Your submission must be the original work of you and your partner, and you must understand everything that is written on your submission.*** *We strongly suggest that you write solutions using LaTex—see the course website for a latex solution template.]*

In this problem set, we define and prove the correctness of the randomized primality testing algorithm of Agrawal and Biswas, that was later derandomized by Agrawal, Kayal, and Saxena. The randomized algorithm is described below, and relies on testing the identity of polynomials over $\mathbb{Z}_n$, modulo a polynomial. As a quick refresher, we define formal polynomials of the variable x modulo an integer n to be of the form $\sum_i c_i x^i$, where the coefficients c_i are integers modulo n. Note that the formal polynomial $x^7 - x \mod 7$ is *not* the same as the zero polynomial despite the fact that if we view this as a *function* over the integers modulo 7, the truth table is identical to that of the zero polynomial (hence two formal polynomials can be distinct even if they are identical when viewed as *functions* over $\mathbb{Z}_7$.

Given a degree d polynomial $p(x)$ with integer coefficients, for any polynomial $q(x)$ with integer coefficients, we say $q(x) \equiv t(x) \mod (p(x), n)$ if there exists some polynomial $s(x)$ such that $q(x) = s(x) * p(x) + t(x) \mod n$. For example $x^5 + 6x^4 + 3x + 1 \equiv 3x + 1 \mod (x^2 + x, 5)$, since $(x^3)(x^2 + x) + (3x + 1) = x^5 + x^4 + 3x + 1 \equiv x^5 + 6x^4 + 3x + 1 \mod 5$.

Algorithm 1. AGRAWAL-BISWAS PRIMALITY TEST

Given n:

- *If n is divisible by 2, 3, 5, 7, 11, or 13, or is a perfect power (i.e. $n = c^r$ for integers c and r) then output* ***composite.***
- *Set d to be the smallest integer greater than $\log n$, and choose a random degree d polynomial with leading coefficient 1:*

$$r(x) = x^d + c_{d-1}x^{d-1} + \ldots + c_1 x + c_0,$$

  by choosing each coefficient c_i uniformly at random from $\{0, 1, \ldots, n-1\}$.
- *If $(x+1)^n \equiv x^n + 1 \mod (r(x), n)$ then output* ***prime,*** *else output* ***composite.***

1. (4 points) In two or three sentences, argue why this algorithm can be run in time poly-logarithmic in n—i.e. time $O\left((\log n)^c\right)$ for some constant c.

2. Now we begin our proof that the algorithm outputs the correct answer with good probability. To start, we prove the polynomial analog of Fermat's Little Theorem, upon which this primality test relies:

(a) (4 points) Prove that if n is prime, then for any integer a, $(x-a)^n = x^n - a \mod n$. [Again, recall the definition of polynomial equivalence modulo n, namely that $\sum_i c_i x^i = \sum_i c'_i x^i \mod n$ if and only if $c_i = c'_i \mod n$ for all i.]

(b) (4 points) Prove that if n is not prime and is not a power of a prime, then for any a s.t. $gcd(a, n) = 1$ and any prime factor p of n, $(x-a)^n \neq x^n - a \mod p$. [Hint: you just need to show that if you were to expand the left side, there is one term other than x^n and $-a$ that does not vanish, modulo p.]

3. (4 points) Given the above polynomial version of Fermat's Little Theorem, why can we not simply use the Schwartz-Zippel randomized test of polynomial identity from Lecture 1 to yield a randomized primality test? (There are at least 2 reasons...it suffices to just give one.)

4. In this part we prove that if n is composite, the probability over random choices of $r(x)$ that the algorithm successfully finds a witness to the compositeness of n is at least $\frac{1}{4d}$.

   (a) (4 points) Using the polynomial version of Fermat's Little Theorem that you proved in part 2, and the fact that, for prime q, every polynomial over $\mathbb{Z}_q$ that has leading coefficient 1 (i.e. "monic") has a unique factorization into irreducible monic polynomials, prove that the number of irreducible degree d factors that the polynomial $(x+1)^n - (x^n+1)$ has over $\mathbb{Z}_p$ is at most n/d, where p is any prime factor of n. (A polynomial is irreducible if it cannot be factored, for example $x^2+1 = (x+1)(x+1) \mod 2$ is not irreducible over $\mathbb{Z}_2$, but x^2+1 is irreducible over $\mathbb{Z}_3$.) [Hint: even though this question sounds complicated, the proof is just one line...don't second-guess yourself :)]

   (b) (4 points) Let $f(d, p)$ denote the number of irreducible monic degree d polynomials over $\mathbb{Z}_p$. Prove that if n is composite, and not a power of a prime, the probability that $r(x)$ is a witness to the compositeness of n is at least $\frac{f(d,p) - n/d}{p^d}$, where p is a prime factor of n. [Hint: p^d is the total number of monic degree d polynomials over $\mathbb{Z}_p$.]

   (c) (4 points) Now complete the proof, and prove that the algorithm succeeds with probability at least $1/(4d)$, leveraging the fact that the number of irreducible monic polynomials of degree d over $\mathbb{Z}_p$ is at least $p^d/d - p^{d/2}$. (You should be able to prove a much better bound, though $1/4d$ is fine.) [Hint: you will also need to leverage the fact that we chose $d > \log n$ and also explicitly made sure that n has no prime factors less than 17.]

   (d) (2 points bonus) Why were we working modulo some prime factor p of n, rather than modulo n itself? Specifically, which of the above parts required this and why?