CS 70 Discrete Mathematics and Probability Theory
Fall 2013 Vazirani Polynomials Practice

1. **Polynomial interpolation**

   (a) Consider the set of four points $\{(0,1),(1,2),(2,4),(4,2)\}$.

   (i) Construct the unique degree-3 polynomial (over the reals) that passes through these four points by writing down and solving a system of linear equations.

   (ii) Repeat part (i) but using the method of Lagrange interpolation. Show your working.

   (b) Find a polynomial $h(x) = ax^2 + bx + c$ of degree at most 2 (over GF(7)) such that $h(0) = 3 \mod 7$, $h(1) = 6 \mod 7$, and $h(2) = 6 \mod 7$.

2. **Roots**

   Let p be prime. Argue that every value z has at most 2 square roots modulo p.

3. **Random Polynomials**

   Let p be a prime. Two polynomials f and g over $GF(p)$ are chosen independently and uniformly at random from all polynomials of degree $d \geq 0$. What is the expected number of intersection points of f and g?

4. **More polynomials!**

   Define the sequence of polynomials by $P_0(x) = x + 12$, $P_1(x) = x^2 - 5x + 5$ and $P_n(x) = xP_{n-2}(x) - P_{n-1}(x)$.
   (For instance, $P_2(x) = 17x - 5$ and $P_3(x) = x^3 - 5x^2 - 12x + 5$.)

   (a) Show that $P_n(7) \equiv 0 \pmod{19}$ for every $n \in \mathbb{N}$.

   (b) Show that, for every prime q, if $P_{2013}(x) \not\equiv 0 \pmod{q}$, then $P_{2013}(x)$ has at most 2013 roots modulo q.

5. **Even more polynomials!**

   Consider two polynomials $p(x), q(x)$ whose product is zero: that is, $p(x) \cdot q(x) = 0$ for all x.

   (a) Show that if $p(x)$ and $q(x)$ are polynomials over the real numbers then in this case, either $p(x) = 0$ for all x or $q(x) = 0$ for all x (or both). (Hint: You may want to first prove this lemma, true in all fields: The set of roots of $p(x) \cdot q(x)$ is the union of the roots of $p(x)$ and $q(x)$.)

   (b) Show that, in contrast, over $GF(p)$ there exist such polynomials whose product is zero but which are both nonzero. (A polynomial $p(x)$ is nonzero if $p(x) \neq 0$ for some, but not necessarily all, x.) (Hint: Fermat's Little Theorem is useful here).

6. **Random Polynomials**

   Recall that a polynomial of degree d has at most d roots. In this problem we will show that most polynomials of degree (at most) d have even fewer roots. We will work modulo p, where p is a prime number such that $d \ll p$.

Let $f(x) = a_d x^d + a_{d-1} x^{d-1} + \cdots + a_0$ be a uniformly random polynomial of degree (at most) d. Recall that $f(x)$ can be picked by either picking each coefficient a_j to be a random number modulo p, or by picking randomly the values of the polynomial at $d+1$ chosen points x and interpolating to get $f(x)$.

Let Z be a random variable whose value is the number of roots of $f(x)$.

(a) What is the probability that $f(x)$ has a root at $x = 1$ (i.e $f(1) = 0$)?

(b) What is E[Z], the expected number of roots of $f(x)$?

(c) What does Markov's bound tell you about the probability that $f(x)$ has at least 10 roots?

(d) What is the probability that $f(x)$ has roots at $x = 1$ and $x = 5$ (i.e. $f(1) = 0$ and $f(5) = 0$)?

(e) Recall that $Var[Z] = E[Z^2] - E[Z]^2$. What is $Var[Z]$?

(f) What does Chebyshev's bound tell you about the probability that $f(x)$ has at least 10 roots?

7. **Secret Sharing**

Consider the following variant of the secret sharing problem. We wish to share a secret among twenty-one people, divided into three groups of seven, so that the following condition is satisfied. A subset of the twenty-one people can recover the secret if and only if it contains majorities (at least four out of seven) of at least two of the groups. How would you modify the standard secret sharing scheme to achieve this condition? (Hint: Try a two-level scheme, one level for groups, the other for people within the group.)

8. **How many secrets?**

A secret sharing scheme is k-secure if and only if any group of k or fewer people has probability at most $\frac{1}{q}$ of recovering the secret, where q is the number of possible choices for the secret (this means that the best strategy such a group has is to guess the secret at random). In the typical secret sharing scheme, the secret is $P(0)$, the value of a certain degree k polynomial (that we construct) at 0. Suppose that, instead, the secret is $P(0), P(1)$ (the values at both 0 and 1). Of course, we also change the algorithm by handing out $P(2), \ldots, P(n+1)$ to the n people instead of handing out $P(1), \ldots, P(n)$. Is this scheme still k-secure? Prover your answer.

9. **Spies**

The president wants to authorize military generals to launch nuclear weapons without the direct approval of the president if enough of them sanction this launch. As you might remember, secret sharing schemes come in handy here. But now there is a new twist. We have been informed that spies have infiltrated the army, and have even become generals. When it is time for a nuclear launch, the spies can give false information and therefore break the usual secret-sharing scheme.

There are 100 generals in the military. The president knows that out of those 100 at most 5 are spies. He wants a secret sharing mechanism where any group of 9 generals cannot launch the nuclear missiles. However he wants n generals to always be able to launch the nuclear missiles, no matter how many of the 5 spies are there among them. What mechanism should the president use? What is the correct bound n that guarantees any n generals can launch the missiles (even if there are spies among them).

10. **Win at Poker**

A *pseudorandom number generator* is a way of generating a large quantity of random-looking numbers, if all we have is a little bit of randomness (known as the *seed*). One simple scheme is the *linear congruential generator*, where we let x_0 denote the seed and define

$$x_{t+1} = (ax_t + b) \bmod m$$

for some modulus m and some constants a, b. (Notice that $0 \leq x_t < m$ holds for every t.)

You've discovered that a popular web site uses a linear congruential generator to generate poker hands for its players. For instance, it uses x_0 to pseudo-randomly pick the first card to go into your hand, x_1 to pseudo-randomly pick the second card to go into your hand, and so on. For extra security, the poker site has kept the parameters a and b secret, but you do know that the modulus is $m = 2^{31} - 1$ (which is prime).

Suppose that you can observe the values x_0, x_1, x_2, x_3, and x_4 from the information available to you, and that the values $x_5, \ldots, x_9$ will be used to pseudo-randomly pick the cards for the next person's hand. Describe how to efficiently predict the values $x_5, \ldots, x_9$, given the values known to you.