

CS 235: Algebraic Algorithms

Final Exam Review Questions

1 Divisibility and Modular Arithmetic

We proved early on in the course that for an odd prime p , half of the values in \mathbb{Z}_p^* have a square root. We did this by showing, first, that every square has at most two roots, and second, that every square has at least two roots, which combined gives us our bound of $1/2$. We will now look at a similar argument for cube roots.

1.1 Simple bound on modular cubes

Using the fact that a degree-3 polynomial has at most 3 zeroes, prove that *at least* $1/3$ of all values have cube roots modulo p .

1.2 Complete bound on modular cubes

Prove that:

- (1) If $p \equiv 1 \pmod{3}$, then *exactly* $1/3$ of the values in \mathbb{Z}_p^* have cube roots.
- (2) Otherwise, *all* values in \mathbb{Z}_p^* have cube roots.

(Hint: Remember that there is always a generator modulo a prime, and think about the relation that powers of that generator would have to cubes and cube roots.)

2 Chinese Remainder

2.1 Primes

Let p and q be distinct primes, and consider the value $(1, 1) \in \mathbb{Z}_p \times \mathbb{Z}_q$. What is the smallest integer i such that $i(1, 1) = (0, 0)$? Express your answer in terms of p and q .

2.2 Composites

What if we have non-prime values m and n ? For $(1, 1) \in \mathbb{Z}_m \times \mathbb{Z}_n$, what is the smallest integer i such that $i(1, 1) = (0, 0)$? Again express your answer in terms of m and n .

3 Solving Polynomials

Let f be a polynomial in $\mathbb{Z}_n[x]$, that is, a polynomial with coefficients in \mathbb{Z}_n , and let n be any integer ≥ 5 . Suppose that there are *exactly* four values a such that $f(a) = 0$.

(1) What is the lowest possible degree of f ? Prove your answer. (2) Give an example showing that it is possible for the degree of f to be *exactly* four. (3) Give an example showing that it is possible for the degree of f to be *greater than* four.

4 Repeated squaring

Given an input $a \in \mathbb{Z}_n$, and using only multiplication and addition, what is the smallest number of arithmetic operations needed to compute:

- (1) a^{256} ?
- (2) a^{255} ?

You can assume all operations are in \mathbb{Z}_n (that is, you don't need to explicitly count computation of the remainders).

5 Subgroups

5.1 Commutative groups

How many subgroups of size 5 are there in \mathbb{Z}_{25} ? In $\mathbb{Z}_5 \times \mathbb{Z}_5$?

5.2 Noncommutative groups

How many subgroups of size 4 are there in D_8 ?

6 Normal subgroups

Remember that a subgroup H of a group G is *normal* if, for all $x \in G$, $xH = Hx$. Give an example of:

- (1) A subgroup of D_8 of size 2 that is *not* normal.
- (2) A subgroup of D_8 of size 2 that *is* normal.

(Hint for this part: D_8 does not have very many subgroups of size 2. Your first step here should be to find them.)

7 Quotient Groups

7.1 Quotients 1

Let $H \subset \mathbb{Z}_{25}$ be a subgroup of size five. What group(s) could \mathbb{Z}_{25}/H be?

7.2 Quotients 2

Let $H \subset \mathbb{Z}_5 \times \mathbb{Z}_5$ be a subgroup of size five. What group(s) could \mathbb{Z}_{25}/H be?

8 Irreducible polynomials

8.1 Degree 2

Prove that $x^2 + x + 1$ is irreducible when coefficients are in \mathbb{Z}_2 , but *not* when they are in \mathbb{Z}_3 .

8.2 Degree 3

List *all* irreducible polynomials of degree 3 with coefficients in \mathbb{Z}_2 .

9 Finite Fields

Consider the finite field of size 8. To be explicit, we will use $\mathbb{Z}_2[x]/(x^3 + x + 1)$. For notational convenience, you may find it best to write field elements as three-bit strings (each

bit corresponding to one coefficient).

9.1 Generators

Find a generator of the field. Prove that it is a generator.

9.2 Computations

Compute the following values:

(1) $001 \cdot 110$

(2) $010 \cdot 010$

(3) $110 \cdot 010$

10 Error-correcting codes

We will now compute Reed-Solomon codes with the finite field from the previous problem. Be careful with notation: though we are working with polynomials in this problem, they are *not* the polynomials used in constructing the field, they are ordinary polynomials whose coefficients and values are taken *from* the field: that is, every input, output, and coefficient of these polynomials is a three-bit string as in the previous problem. To avoid any ambiguity, you may wish to write these polynomials with an upper-case X for the variable (or some other letter entirely).

10.1 Code size

What is the *largest* number of field elements we could possibly transmit with a single polynomial (assuming there are no errors)?

10.2 Encoding

Suppose we decide to work with polynomials of degree ≤ 3 , so we will be transmitting the equivalent of 4 field elements (the coefficients) via a block size of 8 field elements (the outputs of the polynomial).

What is the code corresponding to the message (000, 000, 001, 111)?

10.3 Lagrange Interpolation

Give a polynomial $f(X)$ that is nonzero when $X = 0$, and zero everywhere else.

10.4 Correcting errors

Suppose you receive an encoded message, and after decoding you obtain the polynomial $010X^7 + 100X^6 + \dots$

Assume there is only one error in the message. Which output contains that error?