Lecture 9: Finite fields

Rajat Mittal*

IIT Kanpur

We studied the set $\mathbb{Z}_p$ in detail for number theory. As mentioned earlier, the set can be thought of as another number system where we can add, subtract, multiply and divide. You have already studied rational number system, real number system and complex number system where addition, subtraction, multiplication and division is well defined.

In a very informal sense, we want to say that all sets having these *nice* operations (having addition, subtraction etc.) are *special* and share interesting properties. These special sets, with their nice operations, will be called *fields* and we will study these abstract object and there properties.

Finite fields, whose size is finite, are a very important sub-class of fields for computer science and mathematics ($\mathbb{Z}_p$ is an example). Later in this note, finite fields and its properties will be studied in detail. We will finish with an application of finite fields in error correcting codes.

1 Fields

We have repeatedly said that elements in $\mathbb{Z}_p$ can be added, subtracted, multiplied and divided. Notice that subtraction is inverse of addition and division is the inverse of multiplication. First, let us formulate the meaning of, *addition can be performed over a set S*.

Exercise 1. For example, our formulation should say that addition is possible over $\mathbb{Z}_n$ but not in $\mathbb{Z}_n^*$. Could you now think of a formulation?

One way to capture it is by these two properties. We can say that addition can be *performed* over S if

1. Addition is allowed: for all $a, b \in S$, $a + b \in S$ and $a + b = b + a$.
2. Subtraction is allowed: There exist inverse for every element a called $-a$ and identity $0 \in S$, such that,

$$a + 0 = 0 + a = a \text{ and } a + (-a) = (-a) + a = 0.$$

Clearly addition is possible over $\mathbb{Z}_n, \mathbb{Z}, \mathbb{R}, \mathbb{C}$ (with identity as 0). Multiplication is not possible, why?. We can remove 0 and then multiplication will be possible on the sets $\mathbb{Z}_p$, rational numbers, real number and complex numbers (with 1 as an identity).

Exercise 2. Show that addition is possible over set of all $n \times n$ matrices. What is the identity?

You can think of many other examples, we will get to study them later.

Let us again look at the set $\mathbb{Z}_p$ where addition as well as multiplication (removing 0) is possible. To collect some interesting properties of $\mathbb{Z}_p$,

- Addition can be performed over $\mathbb{Z}_p$,
- Multiplication can be performed over $\mathbb{Z}_p \setminus \{0\}$,
- Distributivity holds, i.e., $a(b + c) = ab + ac$ for all $a, b, c \in \mathbb{Z}_p$.

Since $\mathbb{Z}_p$ had these properties, we were able to do modular arithmetic (define a new number system) over the set $\mathbb{Z}_p$. Notice that these properties are also true for set of rational numbers($\mathbb{Q}$), set of real numbers($\mathbb{R}$) and set of complex numbers($\mathbb{C}$).

Exercise 3. What is wrong with $\mathbb{Z}$?

* Thanks to Nitin Saxena for his notes from the previous iteration of the course.

The properties seem to *abstract out* the reason why $\mathbb{Z}_p$ can be called *special*. Let us take the next step and say that any set with two possible operations in the above sense is special (it will be called a *field*). Formally,

Definition 1. *The set F with the two operations $+$ and $\times$ is a* field*, if,*

- *$+$ can be performed over F. Say 0 is the identity with respect to $+$ in F.*
- *$\times$ can be performed over $F \setminus \{0\}$.*
- *The two operations $+$ and $\times$ follow the distributive law, i.e.,*

$$a \times (b + c) = a \times b + a \times c \text{ and } (a + b) \times c = a \times c + b \times c.$$

For simplicity of notation, we will call the two operations addition and multiplication (order is important). Generally additive identity is denoted as 0 and the multiplicative identity as 1.

Exercise 4. Why are we excluding 0 when taking elements of F under multiplication?

Because of distribution, $a \times (b + 0) = a \times b + a \times 0$, implies $a \times 0 = 0$. Can't have inverse of 0.

Let us look at some of the examples of fields.

- $\mathbb{Z}$ is NOT a field.
- $\mathbb{Q}$, $\mathbb{R}$ and $\mathbb{C}$ are fields.
- $\mathbb{Z}_m$ is a field iff m is a _____. Ex: Fill in the blank.

Exercise 5. Show that if $ab = 0$ in a field than either $a = 0$ or $b = 0$.

One of the important sub case of fields is when they are finite, like the last example. Finite fields have lot of applications in computer science. We will study one, to construct error correcting codes using finite fields.

Let us move on to more properties of fields.

1.1 Characteristic of a field

The characteristic of a field $\mathbb{F}$ is the minimum $n \in \mathbb{N}$, s.t., $n1 = 0$. Here $n1$ denotes the addition of the multiplicative identity n times,

$$\underbrace{1 + 1 + \cdots + 1}_{n \text{ times}}.$$

In general, the characteristic might not exist for field (say $\mathbb{R}$). In that case we say that characteristic is zero. For the case of finite field though, the characteristic is always a positive number.

Exercise 6. Why is the characteristic of a finite field also finite?

Suppose n is a characteristic of a finite field. If n is composite, say $n = pq$, then $(p1)(q1) = 0$. Since $\mathbb{F}$ is a field, from exercise in the previous section, either $p1 = 0$ or $q1 = 0$, establishing contradiction. So we get the theorem,

Theorem 1. *The characteristic of a finite field is always a prime.*

Exercise 7. Show that $p1 = 0$ implies $pf = 0$ for all $f \in \mathbb{F}$.

$$0 = (1 + \cdots + 1 + 1)f = f + \cdots + f + f = fd$$

One of the example of field of characteristic p is the set $\mathbb{Z}_p$ with modular addition and multiplication. We call this field $\mathbb{F}_p$.

Note 1. There are other finite fields with characteristic p but we will not study them in this course.

2 Polynomials over fields

You are used to thinking of a polynomial (like $4x^2 + 2x + 6$) as an expression of coefficients (in $\mathbb{Z}, \mathbb{R}$ etc.) and variables (x, let us worry about only one variable).

The purpose of this section is to give a formal treatment to constructing polynomials and the rules over them. We will re-derive many properties of polynomials with the only assumption, coefficients arise from a field.

A polynomial over a field $\mathbb{F}$ is a formal sum $a_n x^n + \cdots + a_1 x + a_0$, where the coefficients come from the field $\mathbb{F}$. The set of all polynomials (in one variable) over a field $\mathbb{F}$ is denoted by $\mathbb{F}[x]$.

The *degree* of the polynomial is the highest power of x with a non-zero coefficient. We will call a polynomial to be *monic*, if its leading coefficient is 1.

We can define the addition and multiplication over polynomials (in $\mathbb{F}[x]$) so as to match the definitions learned till now.

Given two polynomials $a(x) = a_n x^n + \cdots + a_1 x + a_0$ and $b(x) = b_n x^n + \cdots + b_1 x + b_0$, their sum is defined as,

$$a(x) + b(x) = (a_n + b_n)x^n + \cdots + (a_1 + b_1)x + (a_0 + b_0).$$

Note 2. If the degree of two polynomials is not equal, we can introduce extra zero coefficients in the polynomial with the smaller degree.

For multiplication, given two polynomials $a(x) = a_n x^n + \cdots + a_1 x + a_0$ and $b(x) = b_m x^m + \cdots + b_1 x + b_0$, their product is defined as,

$$p(x) = a(x)b(x) = (a_n b_m)x^{n+m} + \cdots + (a_2 b_0 + a_1 b_1 + a_0 b_2)x^2 + (a_0 b_1 + a_1 b_0)x + (a_0 b_0).$$

More formally, the product is defined using distribution and the fact that $(ax^i)(bx^j) = (ab)x^{i+j}$. Remember that the calculations over coefficients are done in the field $\mathbb{F}$.

Exercise 8. What is the degree of $a(x)b(x)$ if the degree of $a(x)$ is n and $b(x)$ is m?

Exercise 9. Suppose $a(x) = 2x^3 + 2x^2 + 2$ is a polynomial in $\mathbb{F}_3[x]$, what is $a(x)^2$?

After defining addition and multiplication we would like to define division and gcd of polynomials. Notice that we will follow the exact same path as numbers to end up with unique factorization theorem.

Theorem 2. *Division: Given two polynomials $f(x)$ and $g(x)$, there exist two* unique *polynomials called quotient $q(x)$ and remainder $r(x)$, s.t.,*

$$f(x) = q(x)g(x) + r(x).$$

where the degree of $r(x)$ is less than the degree of $g(x)$.

Proof. Existence: Suppose the degree of $f(x)$ is less than degree of $g(x)$ then $q(x) = 0$ and $r(x) = f(x)$. This will be the base case and we will apply induction on the degree of $f(x)$.

Say $f(x) = f_n x^n + \cdots + f_1 x + f_0$ and $g(x) = g_m x^m + \cdots + g_1 x + g_0$ with $m \leq n$. Multiply g by $f_n g_m^{-1} x^{n-m}$ and subtract it from f.

$$f(x) - f_n g_m^{-1} x^{n-m} g(x) = (f_{n-1} - g_{m-1} f_n g_m^{-1})x^{n-m-1} + \cdots = l(x).$$

So l is a polynomial with lower degree and by induction it can be written as $l(x) = q'(x)g(x) + r(x)$. This implies $f(x) = (f_n g_m^{-1} x^{n-m} + q'(x))g(x) + r(x)$. So we can always find $q(x)$ and $r(x)$ with the condition given above. This method is called *long division* and is the usual method of dividing two numbers.

Exercise 10. What is the relation between the usual division between two integers you learnt in elementary classes and long division.

Replace x by 10.

Uniqueness: Suppose there are two such decompositions $f = q_1 g + r_1$ and $f = q_2 g + r_2$ (notice that we have suppressed x for the sake of brevity). Then subtracting one from another,

$$0 = (q_1 - q_2)g + (r_1 - r_2).$$

Exercise 11. Show that this implies q and r are unique.

□

Using division algorithm, we can define Euclidean GCD algorithm.

Let's define *greatest common divisor (gcd)* first. Given two polynomials f, g, their greatest common divisor is the highest degree polynomial which divides both f, g.

The important observation for Euclidean gcd is, if $f = gq_1 + r_1$ then

$$gcd(f, g) = gcd(g, r_1).$$

Without loss of generality we can assume that f has higher degree than g and hence r has lower degree than g and f. We can continue this process, say $g = q_2 r_1 + r_2$. Then the task reduces to finding the gcd of r_1 and r_2. Ultimately we get two polynomials, s.t., $r_n \mid r_{n-1}$. Then r_n is the gcd of f and g. If r_n is a constant, we say that the gcd is 1.

Exercise 12. Show that any polynomial which divides both f and g will also divide r_n mentioned above. Show that r_n divides both f and g.

From the previous exercise it is clear that r_n is one of the gcd (it divides both and has highest degree).

Exercise 13. Is gcd unique?

only up to multiplication by a constant

Exercise 14. Imp: Show that using Euclidean algorithm for gcd, if $gcd(f, g) = d$ then there exist two polynomials p, q, s.t., $d = pf + qg$.

Let's define *primes* in the ring of polynomials, they are called *irreducible* polynomials (irreducible elements of $\mathbb{F}[x]$). A polynomial f is *irreducible* iff it is not constant and there does NOT exist two non-constant polynomials g and h, s.t., $f = gh$.

Exercise 15. Given that a monic polynomial g is irreducible, show, any polynomial f is divisible by g or their gcd is 1. This property can be re-stated, any irreducible polynomial can't have a non-trivial gcd (trivial gcd: 1 or the polynomial itself).

With this definition, we can start finding the factors of any polynomial f. Either f is irreducible or it can be written as gh. If we keep applying this procedure to g and h. We get that any polynomial f can be written as,

$$f = cg_1 g_2 \cdots g_k.$$

Where g_i's are irreducible monic polynomials and c is a constant in the field $\mathbb{F}$.

Can two such factorizations exist? It turns out, like in the case of natural numbers, this factorization is unique up to ordering of polynomials. For the sake of contradiction, suppose there are two such factorizations $cg_1 \cdots g_k$ and $ch_1 \cdots h_l$.

Exercise 16. Why can we assume that the constant is same for both factorizations?

We know that since g_1 is irreducible it can't have a non-trivial gcd with either $h = h_1 \cdots h_{l-1}$ or h_l. We will also show that it can't have gcd 1 with both. Suppose $gcd(h_l, g_1)$ is 1. Then using Euclidean gcd,

$$1 = ph_l + qg_1 \Rightarrow h = pf + qg_1 h.$$

Since g_1 divides both terms on the R.H.S, it divides h. Hence the gcd of h and g_1 is g_1. So g_1 either divides h_l or $h = h_1 \cdots h_{l-1}$.

If it divides $h_1 \cdots h_{l-1}$, we can further divide it and ultimately get that g_1 divides h_i for some i. Since both g_1 and h_i are irreducible and monic, we get $g_1 = h_i$. Cancelling g_1 from both sides,

$$g_1(g_2 \cdots g_k - h_1 \cdots h_{i-1} h_{i+1} \cdots h_l) = 0 \Rightarrow g_2 \cdots g_k - h_1 \cdots h_{i-1} h_{i+1} \cdots h_l = 0.$$

Exercise 17. Prove the above implication.

Continuing the same way, it can be proven that both factorizations are same up to a permutation.

Theorem 3. *Unique factorization: Given a polynomial f it can be written in a unique way as a product of irreducible monic polynomials up to ordering.*

$$f(x) = cg_1(x)g_2(x) \cdots g_k(x)$$

Where c is a constant in the field $\mathbb{F}$ (the leading coefficient of f) and g_i's are irreducible monic polynomials.

Exercise 18. The order of going from division algorithm to Euclidean GCD to unique factorization is important. Where else have you seen this?

There is an easy way to find out whether a degree one polynomial $x - a$ divides a polynomial f or not. Substitute a in the polynomial f (we call the evaluation f(a)), if it evaluates to zero then $x - a$ divides f otherwise not. If $f(a) = 0$, we say that a is a *root* of f. The proof of this is given as an exercise.

Using the factorization theorem we can show that any polynomial of degree d can have at most d roots. The proof of this theorem is left as an exercise.

Theorem 4. *Given a polynomial p of degree d over a field $\mathbb{F}$. There are at most d distinct roots of p.*

3 Cyclic structure of $\mathbb{F}_p$

Observe that every element of $\mathbb{F}_p$ can be obtained by adding 1 enough number of times. Such an element is called an additive generator of $\mathbb{F}_p$. There is a nice cyclic structure to elements of $\mathbb{F}_p$ under addition.

Exercise 19. Show that every element of $\mathbb{F}_p$ is an additive generator except 0.

What about multiplication? If we remove 0, can every other element be obtained as a power of a single element?

Try it for $\mathbb{F}_3$, $\mathbb{F}_5$ and $\mathbb{F}_7$. How many generators are there?

It turns out that every field $\mathbb{F}_p$ has at least one multiplicative generator (actually all finite fields). A multiplicative generator of a field $\mathbb{F}$ is also called a *primitive element* of $\mathbb{F}$.

Before we prove the existence of a primitive element, let us simplify the question.

Look at the consecutive powers of an element $a \in \mathbb{F}_p$. The values $1, a, a^2, \cdots$ need to repeat as all of them are elements of a finite set. Suppose, the first collision is at powers i and j, $a^i = a^j$. This will imply $a^{i-j} = 1$.

The previous argument shows that the first value which repeats is 1, and all values before that are distinct. The first exponent e such that $a^e = 1$ is called the *order* of the element a in $\mathbb{F}_p$. So, the consecutive powers of a look like,

$$1, a, a^2 \cdots a^e = 1, a^{e+1} = a, a^{e+2} = a^2, \cdots$$

Exercise 20. Let e be the order of a. Show that $a^n = 1$ iff $e \mid n$.

Exercise 21. Suppose the order of a in a group G is d. Show that for element a^k, order is $\frac{d}{gcd(d,k)}$.

If all elements, except 0, can be represented as powers of a, then the order of a should be $p-1$. Conversely, if the order is $p-1$ then a generates every element except 0. Why?

So, we need to show that for all p, there exist an a whose order is $p-1$ (a is a primitive element).

Theorem 5. *For any prime p, there exist $\phi(p-1)$ primitive elements of $\mathbb{F}_p$.*

Proof. Let's call the multiplicative part of our field to be $F^* = \mathbb{Z}_p \setminus \{0\}$, then $|F^*| = p-1$. Since, for all elements x of F^*,

$$x^{p-1} - 1 = 0$$

So, there are exactly $p-1$ roots of the above equation (why exactly $p-1$?).

For any element x, the order d divides $p-1$, hence x is a solution of $p(d) = x^d - 1$ for some $d \mid (p-1)$. Notice that the polynomial $p(d)$ has at most d roots.

For the sake of contradiction, suppose there are no primitive elements. Then, every element has order strictly less than $p-1$. We would like to show that there are not enough roots $(p-1)$ for the polynomial $x^{p-1} - 1$.

So we would like to show,

$$\sum_{d<(p-1), d|(p-1)} d < p-1 \tag{1}$$

Note 3. There is a strict inequality $d < p-1$ in the summation index as well as the inequality.

Exercise 22. Show that this is not true for some p.

The reason why the strategy above does not work is, we are counting lot of elements multiple times. A solution of $p(d)$ will be a solution of $p(2d), p(3d), \cdots$. There is a decent chance that some of numbers $2d, 3d, \cdots$ might be divisors of $p-1$ too.

So, say $e(d)$ is the number of elements with order *exactly* d. Hence, instead of Eq. 1, the contradiction will be shown by proving the equation,

$$\sum_{d<p-1, d|(p-1)} e(d) < p-1 \tag{2}$$

This equation follows from the following two claims. The proof of first one is left as an exercise, other will be proved here.

Note 4. $\phi(d)$ is the number of elements co-prime to d.

Claim. For a number n, $\sum_{d|n} \phi(d) = n$.

Proof hint: For any number $k \leq n$, look at $gcd(k,n)$ and $\frac{k}{gcd(k,n)}$.

Claim. If there exist an element of order d then $\phi(d) = e(d)$.

Proof. Suppose, the element with order d is x. Then, d roots for $x^d - 1$ are precisely $x^0, x^1, \cdots, x^{d-1}$ (these are the d roots and there are at most d roots). The order of x^k is $\frac{d}{gcd(d,k)}$.

Hence, the elements with order d are precisely x^k, s.t., $gcd(d,k) = 1$. So $e(d) = \phi(d)$.

□

Using the claims,

$$p-1 = \sum_{d|(p-1), d \leq p-1} \phi(d) > \sum_{d|(p-1), d<p-1} e(d). \tag{3}$$

The inequality follows because $e(d) \leq \phi(d)$ and we have assumed $e(p-1) = 0$. So, the equation 2 follows from non-existence of primitive element and we get a contradiction.

By definition of $e(d)$, $\sum_{d|(p-1)} e(d) = p-1$. Hence, there should be equality in the second part of Eq. 3. That means, there are exactly $\phi(d)$ elements of order d in a field $\mathbb{F}_p$ where $d \mid p-1$. Specifically, there are $\phi(p-1)$ primitive elements for the field $\mathbb{F}_p$.

□

4 Error correcting codes

We will look at another application, constructing error correcting codes. Again, the problem is similar to cryptography, where we want to transmit a message on an unreliable network. The difference is, the network is noisy and not unsecure. In other words, there is no adversary (Eve) but random changes in the message can happen due to the noise in the environment. Instead of encrypting the data into ciphertext and decrypting it, we will encode the message into a codeword and decode it back into the message.

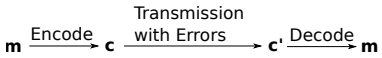

Fig. 1. Error correction

4.1 Basics of error correcting codes

The message m for the setting will be considered as a string of length k over an alphabet Σ. So, m will be an element of Σ^k. This message will be encoded into a codeword c of a bigger length, it will be an element of Σ^n.

When transmitted over a noisy channel, some letters of c might be changed (we will assume a bound on it, otherwise coding won't be possible) and we will obtain a faulty codeword c'. The task is to design a coding mechanism, such that, m can be obtained even from c'.

Note 5. The recovery will only be possible if only a few letters of c gets changed.

In a nutshell, coding is an injective function $E : \Sigma^k \to \Sigma^n$. Given a c' which is *close to* $E(m)$ (say, only t letters changed), we can recover m efficiently. The set of all codewords, of total size $|\Sigma|^k$, is called the code C.

Exercise 23. Can be function E be surjective?

One of the easiest choice for a code is the repetition code, you just send the message multiple times and take the majority output. Though n will be really large in this case. Our job is to design *better* codes.

The natural choice for alphabet is the binary one, $\{0, 1\}$, which can be also be thought of as $\mathbb{F}_2$. We will assume that the alphabet Σ is equal to $\mathbb{F}_p$ for some prime p. So, encoding is a mapping $E : \mathbb{F}_p^k \to \mathbb{F}_p^n$.

Note 6. This provides us with the flexibility of adding two codewords place-wise.

To formalize the notion of distance between two elements of Σ^n (c' and c mostly), we define *Hamming distance*. The Hamming distance between c and c' is the number of places where they differ. To give an example, if t letters get changed under transmission, then the Hamming distance between c and c' is t.

Exercise 24. Consider $\mathbb{F}_7$, what is the Hamming distance between 123456 and 132456. What about 112256 and 221165?

2 and 6

The *distance* of a code C is defined to be the minimum hamming distance between any two codewords of the code C. If the distance is d with k length string going to n length string, it is called a (n, k, d) code.

Exercise 25. Why is the distance important?

Suppose the distance of C is large, more than 100. If we know that the transmission changes at most 50 places, then we can recover the message. For every faulty codeword c' there is at most one codeword $c \in C$.

Exercise 26. Why is this true?

If there are two valid codewords with Hamming distance at most 50 from c' then the Hamming distance between them is less than equal to 100.

This shows that if the distance is d, we can correct errors of size less than $d/2$. Though, this might require lot of time and space (keeping the elements near to codewords). We generally look for codes which have efficient coding and decoding procedure.

In any case, we need to find codes with large distance. What can we achieve?

Exercise 27. Given n what is the best distance possible? What is the best k possible?

From the previous exercise, it is clear that there is a trade-off between k and d. Given n, we want to keep both k and d high. Though, it seems that increasing one decreases other. Let us look at the best possible d given n and k in an (n, k, d) code.

Theorem 6 (Singleton bound). *For any (n, k, d) code, $d \leq n - k + 1$.*

Proof. If we look at the first $k-1$ coordinates of every codeword, at least two codewords will have identical first $k-1$ letters (Pigeonhole principle). These two codewords can have distance at most $n-(k-1)$, proving the bound. □

We will see a construction in next section where $d = n - k + 1$ based on finite fields. Not just that, it also allows us to encode and decode easily.

4.2 Reed-Solomon codes

The Reed-Solomon codes were given in 1960 and used the polynomials defined over finite fields. Before we show the codes, let us look at a fundamental property of polynomials.

Exercise 28. Can two polynomials, both of degree less than k, have same evaluations at k distinct points?

No. The difference of these two polynomials will be a polynomial with degree less than k but more than k roots.

Interpolation: Given k distinct elements $x_1, \cdots, x_k$ and their evaluations $y_1, \cdots, y_k$; Interpolation constructs a polynomial p, such that,

$$p(x_i) = y_i \quad \forall\ i \in \{1, 2, \cdots, k\}.$$

The idea is,

- construct a polynomial which takes value 1 at say x_1 and 0 at every other x_i,
- then combine k such polynomials.

Exercise 29. Can you find a polynomial of degree less than k which has value 1 at x_1 and 0 on all other x_i's?

The following polynomial achieves this,

$$p_1(x) = \Pi_{i \neq 1} \left(\frac{x - x_i}{x_1 - x_i} \right).$$

This allows us to construct the polynomial with $p(x_i) = y_i$ for all i.

$$p(x) = \sum_{i=1}^{k} y_i p_i(x).$$

You should convince yourself that $p(x)$ satisfies the required conditions.

Construction: Every message $m = a_0 a_1 \cdots a_{k-1} \in \mathbb{F}_p^k$ is thought of as a polynomial $m(x) = a_0 + a_1 x + \cdots + a_{k-1} x^{k-1}$ of degree less than k. The codeword for m is simply the evaluation of $m(x)$ at n distinct points of the finite field,

$$c = m(\beta_1) m(\beta_2) \cdots m(\beta_n).$$

Two such codewords can't agree at more than $k - 1$ positions, otherwise there will be two polynomials of degree less than k with exact same evaluations at k distinct points. This implies that the distance d of the code is $d = n - k + 1$.

Encoding requires us to evaluate the polynomial at all points of the finite field and is efficient. Decoding is again based on the properties of these polynomials and linear algebra. We strongly encourage the students to read about decoding on their own.

References

1. K. H. Rosen. Discrete Mathematics and Its Applications. *McGraw-Hill*, 1999.
2. N. L. Biggs. Discrete Mathematics. *Oxford University Press*, 2003.