CS70: Discrete Mathematics and Probability Theory

UC Berkeley

Kelvin Lee

October 14, 2020

Contents

1 Graph Theory

2 Modular Arithmetic

2.1 Congruence

Definition 1 (Congruence). x is **congruent** to y modulo m or $x \equiv y \pmod{m}$ if and only if any one of the following is true:

- $(x - y)$ is divisible by m
- x and y have the same remainder w.r.t. m
- $x = y + km$ for some integer k
- In modulo m, only the numbers $\{0, 1, 2, \ldots, m-1\}$ exist.
- Division is not well-defined.

2.2 Multiplicative Inverse

Definition 2 (Multiplicative Inverse). In the modular space, the **multiplicative inverse** of $x \bmod m$ is y if $xy \equiv 1(\text{mod} m)$.

Theorem 3 (Modular operations). $a \equiv c \bmod m$ and $b \equiv d \bmod m \implies a + b \equiv c + d \ (\text{mod} m)$ and $a \cdot b \equiv c \cdot d \ (\text{mod} m)$.

Theorem 4 (Existence of multiplicative inverse). gcd $(x, m) = 1 \implies x$ has a multiplicative inverse modulo m and it is **unique**.

2.3 Euclid's Algorithm

Question. How do we compute gcd of two numbers x and y?

Theorem 5 (Euclid's Algorithm). Let $x \geq y > 0$. Then

$$gcd(x, y) = gcd(y, x \bmod y)$$

Example 2.1. Compute gcd(16,10):

$$\begin{aligned} \text{gcd } (16, 10) &= \text{gcd } (10, 6) \\ &= \text{gcd } (6, 4) \\ &= \text{gcd } (4, 2) \\ &= \text{gcd } (2, 0) \\ &= 2. \end{aligned}$$

2.4 Extended Euclid's algorithm

Question. How to compute the multiplicative inverse?

- Need an algorithm that returns integers a and b such that:

$$\gcd(x, y) = ax + by.$$

Theorem 6 (Bézout's Identity)**.** For nonzero integers x and y, let d be the greatest common divisor such that $d = \gcd(x, y)$. Then, there exist integers a and b such that

$$ax + by = d.$$

- When $\gcd(x, y) = 1$, we can deduce that b is an inverse of y mod x.
- This uses back substitutions repetitively so that the final expression is in terms of x and y.

2.5 Functions

Definition 7 (Function)**.** Let A and B be nonempty sets. A **function** f from A to B is an assignment of exactly one element of B to each element of A. (vertical line test)

- To denote such a function, we write $f : A \to B$ (f maps A to B).
- A is the **domain** and B is the **co-domain**.
- Pre-image is a **subset** of domain, and the image/range is the **subset** of co-domain.
  - If $f(a) = b$, where $a \in A$ and $b \in B$, then we say that b is the image of a and a is the pre-image of b.

2.6 Bijection

Definition 8 (One-to-one)**.** A function f is said to be **one-to-one** if and only if $f(a) = f(a')$ implies that $a = a'$ for all $a, a' \in A$. A function is said to be **injective** if it is **one-to-one**.

- To show that a function is *one-to-one*, we show that $a \neq a' \implies f(a) \neq f(a')$. (Why?)

Definition 9 (Onto)**.** A function f is called **onto**, or a surjection, if and only if for every element $b \in B$ there is an element $a \in A$ such that $f(a) = b$. We also say that f is **surjective** if it's onto.

- To show that a function is *onto*, choose $a = f^{-1}(b)$ and so $f(f^{-1}(b)) = b$.

Definition 10 (Bijection)**.** A function f is a **bijection** if and only if it is both *one-to-one* and *onto*. We also say that f is bijective.

- If $f : A \to B$ is a bijection, it will have an **inverse** function (a lemma from notes), and $|A| = |B|$.

2.7 Fermat's Little Theorem

Theorem 11 (Fermat's Little Theorem)**.** For any prime p and any $a \in \{1, 2, ..., p-1\}$, we have
$$a^{p-1} \equiv 1 \pmod{p}.$$

Proof. Consider $S = \{1, 2, \ldots, p-1\}$ and $S' = \{a \bmod p, 2a \bmod p, \ldots, (p-1)a \bmod p\}$. They are the same set under mod p (different order).

$$\prod_{k=1}^{p-1} k \equiv \prod_{k=1}^{p-1} ka \pmod{p}$$
$$(p-1)! \equiv a^{p-1}(p-1)! \pmod{p}$$
$$a^{p-1} \equiv 1 \pmod{p}$$

□

2.8 Chinese Remainder Theorem

Theorem 12 (Chinese Remainder Theorem)**.** Let $n_1, n_2, \ldots, n_k$ be positive integers that are coprime to each other. Then, for any integers a_i, the system of simultaneous congruences

$$x \equiv a_1 \pmod{n_1}, x \equiv a_2 \pmod{n_2}, \ \ldots \ , x \equiv a_k \pmod{n_k}$$

has a unique solution
$$x = \left(\sum_{i=1}^{k} a_i b_i\right) \bmod N$$

where $N = \prod_{i=1}^{k} n_i$ and $b_i = \frac{N}{n_i}\left(\frac{N}{n_i}\right)_{n_i}^{-1}$ where $\left(\frac{N}{n_i}\right)_{n_i}^{-1}$ denotes the multiplicative inverse $(\bmod n_i)$ of the integer $\frac{N}{n_i}$.

Proof. To see why x is a solution, notice that for each $i = 1, 2, \ldots, k$, we have

$$\begin{aligned} x &\equiv a_1 y_1 z_1 + a_2 y_2 z_2 + \cdots + a_k y_k z_k && \pmod{n_i} \\ &\equiv a_i y_i z_i && \pmod{n_i} \\ &\equiv a_i && \pmod{n_i}. \end{aligned}$$

- The second line follows since $y_j \equiv 0 \bmod n_i$ for each $j \neq i$.
- The third line follows since $y_i z_i \equiv 1 \bmod n_i$.

Now, to prove uniqueness, suppose there are two solutions x and y.

- Then $n_1 \,|(x-y), n_2|\,(x-y), \ldots, n_k \mid (x-y)$.
- Since $n_1, n_2, \ldots, n_k$ are relatively prime, we have that $n_1 n_2 \cdots n_k$ divides $x-y$, or

$$x \equiv y \pmod{N}.$$

Thus, the solution is unique modulo N. □

General construction:

1. Compute $N = n_1 \times n_2 \times \cdots \times n_k$.
2. For each $i = 1, 2, \dots, k$, compute
$$y_i = \frac{N}{n_i} = n_1 n_2 \cdots n_{i-1} n_{i+1} \cdots n_k.$$
3. For each $i = 1, 2, \dots, k$, compute $z_i \equiv y_i^{-1} \bmod n_i$ (z_i exists since $n_1, n_2, \dots, n_k$ are pairwise coprime).
4. Compute
$$x = \sum_{i=1}^{k} a_i y_i z_i$$
and $x \bmod N$ is the unique solution modulo N.

Intuitive way to solve for CRT:

1. Begin with the congruence with the largest modulus, $x \equiv a_k \pmod{n_k}$.
2. Re-write this modulus as an equation, $x = j_k n_k + a_k$, for some positive integer j_k.
3. Substitute the expression for x into the congruence with the next largest modulus, $x \equiv a_k \pmod{n_k} \implies j_k n_k + a_k \equiv a_{k-1} \pmod{n_{k-1}}$.
4. Solve this congruence for j_k.
5. Write the solved congruence as an equation, and then substitute this expression for j_k into the equation for x.
6. Continue substituting and solving congruences until the equation for x implies the solution to the system of congruences.

Example 2.2. Solve for the following system of congruences
$$\begin{cases} x \equiv 1 & \pmod 3 \\ x \equiv 4 & \pmod 5 \\ x \equiv 6 & \pmod 7 \end{cases}$$

Solution. Start with mod 7.

1. Write $x = 7k + 6$.
2. Then we have $7k + 6 \equiv 4 \pmod 5 \implies k \equiv 4 \pmod 5$.
3. Then solving for k gives $5j + 4$.
4. Now we have $x = 7k + 6 = 7(5j + 4) + 6 = 35j + 34$.
5. Then $35j + 34 \equiv 1 \pmod 3 \implies j \equiv 0 \pmod 3 \implies j = 3t$.
6. Finally, we have $x = 35(3t) + 34 = 105t + 34 \implies x \equiv \boxed{34} \pmod{105}$.

3 RSA

3.1 Basic Ideas

- Alice and Bob wish to communicate secretly over some (insecure) link, and Eve tries to discover what they are saying.
- Alice transmits a message x (in binary) to Bob by applying her **encryption function** E to x and send the encrypted message $E(x)$ over the link.
- Bob, after receiving $E(x)$, applies his **decryption function** D to it and recover the original message: i.e., $D(E(x)) = x$.
- Since the link is insecure, Eve may know what $E(x)$ is.
- We would like to have an encryption function E such that only knowing $E(x)$ cannot reveal anything about x.
- The idea is that each person has a **public key** known to the whole world and a **private key** known only to him- or herself.
- Alice encodes x using Bob's public key. Bob then decrypts it using his private key, thus retrieving x.

3.2 RSA Scheme

- Let p and q be two large primes, and let $N = pq$ (p and q are not public).
- Treat messages to Bob as numbers modulo N, excluding trivial values 0 and 1.
- Let e be any number that is relatively prime to $(p-1)(q-1)$ (Typically e is a small value).
- Then Bob's public key is the pair of numbers (N, e) and his private key is $d = e^{-1} \pmod{(p-1)(q-1)}$.

3.3 RSA Encryption

- **Encryption**: Alice computes the value $E(x) = x^e \bmod N$ and sends this to Bob.
- **Decryption:** Upon receiving the value $y = E(x)$, Bob computes $D(y) = y^d \bmod N$; this will be equal to the original message x.

Theorem 13. Using the encryption and decryption functions E and D, we have $D(E(x)) = x \pmod N$ for every possible message $x \in \{0, 1, ..., N-1\}$.

Proof. This can be proved using Chinese Remainder Theorem or Fermat's Little Theorem. For more details, please refer to notes. □

4 Polynomials

4.1 Properies of polynomials

- **Property 1:** A non-zero polynomial of degree d has at most d roots.
- **Property 2:** A polynomial of degree d is **uniquely** determined by $d+1$ distinct points.

4.2 Polynomial Interpolation

Question. Given $d+1$ distinct points, how do we determine the polynomial?

- We use a method called **Lagrange Interpolation**, which works similarly to the **Chinese Remainder Theorem**.
- Suppose the given points are $(x_1, y_1), \ldots, (x_{d+1}, y_{d+1})$. We want to find a polynomial $p(x)$ such that $p(x_i) = y_i$ for $i = 1, \ldots, d+1$.
- In other words, we want to find polynomials $p_1(x), \ldots, p_{d+1}(x)$ such that

$$\begin{aligned}
&p_1(x) = 1 \text{ at } x_1 \text{ and } p_1(x) = 0 \text{ at } x_2, \ldots, x_{d+1};\\
&p_2(x) = 1 \text{ at } x_2 \text{ and } p_2(x) = 0 \text{ at } x_1, x_3 \ldots, x_{d+1};\\
&p_3(x) = 1 \text{ at } x_3 \text{ and } p_3(x) = 0 \text{ at } x_1, x_2, x_4, \ldots, x_{d+1} \text{ and so on...}
\end{aligned}$$

4.3 Lagrange Interpolation

- Let's start by finding $p_1(x)$.
- Since $p_1(x) = 0$ at $x_2, \ldots, x_{d+1}$, $p_1(x)$ must be a multiple of

$$q_1(x) = (x - x_2)(x - x_3) \ldots (x - x_{d+1}).$$

- We also need $p_1(x) = 1$ at x_1. Notice that

$$q_1(x_1) = (x_1 - x_2)(x_1 - x_3) \ldots (x - x_{d+1}).$$

- Then $p_1(x) = \frac{q_1(x)}{q_1(x_1)}$ is the polynomial we are looking for.
- Similarly for $p_i(x)$, we have $p_i(x) = \frac{q_i(x)}{q_i(x_i)}$.
- After finding $p_1(x), \ldots, p_{d+1}(x)$, we can construct $p(x)$ by scaling up each bit by corresponding y_i:

$$p(x) = \sum_{i=1}^{d+1} y_i \cdot p_i(x)$$

  This should remind you of CRT.

- Now let us define $\Delta_i(x)$ in the following way (think of them as a basis):

$$\Delta_i(x) = \frac{\prod_{i \neq j}(x - x_j)}{\prod_{i \neq j}(x_i - x_j)}.$$

- Then we have an **unique** polynomial

$$\boxed{p(x) = \sum_{i=1}^{d+1} y_i \Delta_i(x).}$$

4.4 Finite Fields

- The properties of a polynomial would not hold if the values are restricted to being natural numbers or integers because dividing two integers does not generally result in an integer.
- However, if we work with numbers modulo m where m is a prime number, then we can add, subtract, multiply and divide.
- Then **Property 1** and **Property 2** hold if the coefficients and the variable x are restricted to take on values modulo m. When we work with numbers modulo m, we are working over a **finite field**, denoted by $GF(m)$ (**Galois Field**).

4.5 Secret Sharing

4.5.1 Basic Ideas

- Suppose there are n people. Let s be the secret number and q be a prime number greater than n and s. We will work over $GF(q)$.
- Pick a random polynomial $P(x)$ of degree $k-1$ such that $P(0) = s$.
- Distribute $P(1), \ldots P(n)$ to each person so that each one receives one value.
- Then in order to know what s is, at least k of the n people must work together so that they can perform **Lagrange interpolation** and find P.
- If there are less than k people, they will learn nothing about s!

5 Error Correcting Codes

5.1 Basic Ideas

- **Goal:** Transmit messages across an **unreliable** communication channel.
- The channel may cause **packets**(parts of the message) to be **lost**, or even **corrupted**.
- **Error correcting code** is an encoding scheme to protect messages against these errors by introducing redundancy.

5.2 Erasure Errors

- **Erasure errors** refer to some packets being **lost** during transmission.
- Suppose that the message consists of n packets and at most k packets are lost during transmission.
- To prevent this error, we encode the initial message into a redundant encoding consisting of $n+k$ packets such that the receiver can reconstruct the message from any n received packets using **Lagrange interpolation**.

5.3 General Errors

- Now suppose the packets are **corrupted** during transmission due to channel noise. Such error is called **general errors**.
- Suppose that k out of n characters are corrupted and we have no idea which k these are.
- To guard against k general errors, we must transmit $n+2k$ characters.
- To reconstruct the polynomial, we need to find a polynomial $P(x)$ of degree $n-1$ such that $P(i) = r_i$ for at least $n+k$ values of i.

5.4 Error-locator Polynomial

- To efficiently find the polynomial $P(x)$, we need the locations of the k errors.
- Let $e_1, ..., e_k$ be the k locations at which errors occurred. We don't know where these errors are.
- Guessing where the errors are will take exponential time, which is inefficient, so we use the **error-locator polynomial**:

$$E(x) = (x - e_1)(x - e_2) \dots (x - e_k).$$

- Then we have the following:

$$P(i)E(i) = r_i E(i) \quad \text{for } 1 \le i \le n+2k.$$

  This is known as the **Berlekamp–Welch algorithm**.

5.5 Berlekamp–Welch algorithm

- Define $Q(x) = P(x)E(x)$. We have $n + 2k$ equations with $n + 2k$ unknown coefficients:
$$Q(i) = r_i E(i) \quad \text{for } 1 \leq i \leq n + 2k.$$
- We can solve the systems of linear equations and get $E(x)$ and $Q(x)$.
- Finally we compute $\frac{Q(x)}{E(x)}$ to obtain $P(x)$.

6 Counting

6.1 Counting Rules

Theorem 14 (First Rule of Counting)**.** If there are n ways of doing something, and m ways of doing another thing after that, then there are $n \times m$ ways to perform both of these actions.

- Order matters(permutations).
- Sampling k elements from n items:
  - With replacement: n^k.
  - Without replacement: $\frac{n!}{(n-k)!}$.

Theorem 15 (Second Rule of Counting)**.** If order doesn't matter count ordered objects and then divide by number of orderings.

- Without replacement and ordering doesn't matter (combinations).
- Number of ways of choosing k-element subsets out of a set of size n:

$$\binom{n}{k} = \frac{n!}{(n-k)!k!}.$$

6.2 Stars and Bars

Stars and Bars is a technique used to solve for problems that sample with replacement but order doesn't matter by establishing a bijection between the problem and the stars and bars problem.

Problem 1. Consider the equation $a+b+c+d = 12$ where a, b, c, d are non-negative integers. How many solutions are there to this equation?

- Let's simplify this problem a little bit. Suppose we have 12 and 3 bars.

$$\star\star\,|\,\star\star|\,\star\star\star\,|\,\star\star\star\star\star$$

- How many ways can we arrange them? $\binom{12+3}{3} = \binom{15}{3}$
- This is the answer to our original problem! Do you see the bijection between the two problems?

Theorem 16 (Stars and Bars)**.** The number of ways to distribute n indistinguishable objects into k distinguishable bins is

$$\binom{n+k-1}{k-1}.$$

- Don't memorize the formula! Try to visualize the problem by connecting it to stars and bars. Draw out the stars and the bars!

- Again, this method is useful for with replacement but order doesn't matter type of problems.

Theorem 17 (Zeroth Rule of Counting:). If a set A has a bijection relationship with a set B, then $|A| = |B|$.

The stars and bars method relies on this counting rule and this is the key to many combinatorial arguments as we will explore further later.

6.3 Binomial Theorem

Theorem 18 (Binomial Theorem). For all $n \in \mathbb{N}$,

$$(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} a^k b^{n-k}.$$

Proof. See notes. □

Corollary 19. For all $n \in \mathbb{N}$,

$$\sum_{k=0}^{n} (-1)^k \binom{n}{k} = 0.$$

Proof. Plug in $a = -1$ and $b = 1$ for the binomial theorem. □

6.4 Combinatorial Proofs

- Intuitive counting arguments. No tedious algebraic manipulation.
- Proofs by stories: same story from multiple perspectives.
- Proving an identity by counting the same thing in two different ways.
- Useful identity:

$$\binom{n}{k} = \binom{n}{n-k}.$$

- Choosing k objects to include is equivalent to choosing $n-k$ objects to exclude.

Example 6.1. Using combinatorial arguments, show that

$$\sum_{i=0}^{n} \binom{n}{i} = 2^n.$$

Proof. We can use binomial theorem by letting $a = b = 1$, however this is not what the question is asking for.
RHS: Total number of subsets of a set of size n.
LHS: The number of ways to choose a subset of size i is $\binom{n}{i}$. To find the total number of subsets, we simply add all the cases when $i = 0, 1, 2, \ldots, n$. □

6.5 Principle of Inclusion-Exclusion

Theorem 20 (Principle of Inclusion-Exclusion(General):)**.** Let $A_1, \ldots, A_n$ be arbitrary subsets of the same finite set A. Then,

$$|A_1 \cup \cdots \cup A_n| = \sum_{k=1}^{n} (-1)^{k-1} \sum_{S \subseteq \{1,\ldots,n\} : |S| = k} |\cap_{i \in S} A_i| .$$

Proof. See notes. □

Theorem 21 (Principle of Inclusion-Exclusion(Simplified):)**.**

$$|A \cup B| = |A| + |B| - |A \cap B|.$$

6.6 Summary

	with replacement	w/o replacement
order matters	n^k	$\frac{n!}{(n-k)!}$
order doesn't matter	$\binom{n+k-1}{k-1}$	$\binom{n}{k}$