Review

- What does it mean for two ints $a, b$ to be congruent mod $m$?
- What is the Division theorem?
- If $a | b$ and $a | c$, does it mean $b | c$?
- What is the Fundamental Theorem of Arithmetic?

Computing GCDs

- Simple algorithm to compute gcd of $a, b$:
  - Factorize $a$ as $p_1^{a_1} p_2^{a_2} \ldots p_n^{a_n}$
  - Factorize $b$ as $p_1^{b_1} p_2^{b_2} \ldots p_n^{b_n}$
  - $\text{gcd}(a, b) = p_1^{\min(a_1, b_1)} p_2^{\min(a_2, b_2)} \ldots p_n^{\min(a_n, b_n)}$
- But this algorithm is not good because prime factorization is computationally expensive! (not polynomial time)
- Much more efficient algorithm to compute gcd, called the Euclidian algorithm

Insight Behind Euclid’s Algorithm

- Theorem: Let $a = bq + r$. Then, $\text{gcd}(a, b) = \text{gcd}(b, r)$
  - e.g., Consider $a = 12, b = 8$ and $a = 12, b = 5$
  - Proof: We’ll show that $a, b$ and $b, r$ have the same common divisors – implies they have the same gcd.
  - Suppose $d$ is a common divisor of $a, b$, i.e., $d | a$ and $d | b$
  - By theorem we proved earlier, this implies $d | a - bq$
  - Since $a - bq = r$, $d | r$. Hence $d$ is common divisor of $b, r$.
  - Now, suppose $d | b$ and $d | r$. Then, $d | bq + r$
  - Hence, $d | a$ and $d$ is common divisor of $a, b$

Using this Theorem

- Theorem: Let $a = bq + r$. Then, $\text{gcd}(a, b) = \text{gcd}(b, r)$
  - Suggests following recursive strategy to compute $\text{gcd}(a, b)$:
    - Base case: If $b$ is 0, then gcd is $a$
    - Recursive case: Compute $\text{gcd}(b, a \mod b)$
  - Claim: We’ll eventually hit base case – why?

Euclidian Algorithm

- Find gcd of 72 and 20
  - $12 = 72 \div 6$
  - $8 = 20 \div 4$
  - $4 = 12 \div 3$
  - $0 = 4 \div 1$
  - gcd is 4!
GCD as Linear Combination

- \( \gcd(a, b) \) can be expressed as a linear combination of \( a \) and \( b \)
- **Theorem**: If \( a \) and \( b \) are positive integers, then there exist integers \( s \) and \( t \) such that:
  \[
  \gcd(a, b) = s \cdot a + t \cdot b
  \]
- Furthermore, Euclidean algorithm gives us a way to compute these integers \( s \) and \( t \) (known as extended Euclidean algorithm)

Example

- Express \( \gcd(72, 20) \) as a linear combination of 72 and 20
- First apply Euclid’s algorithm (write \( a = bq + r \) at each step):
  1. \( 72 = 3 \cdot 20 + 12 \)
  2. \( 20 = 1 \cdot 12 + 8 \)
  3. \( 12 = 1 \cdot 8 + 4 \)
  4. \( 8 = 2 \cdot 4 + 0 \Rightarrow \gcd \text{ is } 4 \)
- Now, using (3), write 4 as \( 12 - 1 \cdot 8 \)
- Using (2), write 4 as \( 12 - 1 \cdot (20 - 1 \cdot 12) = 2 \cdot 12 - 1 \cdot 20 \)
- Using (1), we have \( 12 = 72 - 3 \cdot 20 \), thus:
  \[
  4 = 2 \cdot (72 - 3 \cdot 20) - 1 \cdot 20 = 2 \cdot 72 + (-7) \cdot 20
  \]

Exercise

Use the extended Euclid algorithm to compute \( \gcd(38, 16) \).

A Useful Result

- **Lemma**: If \( a, b \) are relatively prime and \( a | bc \), then \( a | c \).
- **Proof**: Since \( a, b \) are relatively prime \( \gcd(a, b) = 1 \)
  - By previous theorem, there exists \( s, t \) such that \( 1 = s \cdot a + t \cdot b \)
  - Multiply both sides by \( c: c = csa + ctb \)
  - By earlier theorem, since \( a | bc, a | ctb \)
  - Also, by earlier theorem, \( a | csa \)
  - Therefore, \( a | csa + ctb \), which implies \( a | c \) since \( c = csa + ctb \)

Question

- Suppose \( ca \equiv cb \pmod{m} \). Does this imply \( a \equiv b \pmod{m} \)?
Another Useful Result

- **Theorem:** If \( ca \equiv cb \pmod{m} \) and \( \gcd(c, m) = 1 \), then \( a \equiv b \pmod{m} \).

Examples

- If \( 15x \equiv 15y \pmod{4} \), is \( x \equiv y \pmod{4} \)?
- If \( 8x \equiv 8y \pmod{4} \), is \( x \equiv y \pmod{4} \)?

Linear Congruences

- A congruence of the form \( ax \equiv b \pmod{m} \) where \( a, b, m \) are integers and \( x \) a variable is called a linear congruence.
- Given such a linear congruence, often need to answer:
  1. Are there any solutions?
  2. What are the solutions?
- **Example:** Does \( 8x \equiv 2 \pmod{4} \) have any solutions?
- **Example:** Does \( 8x \equiv 2 \pmod{7} \) have any solutions?
- **Question:** Is there a systematic way to solve linear congruences?

Determining Existence of Solutions

- **Theorem:** The linear congruence \( ax \equiv b \pmod{m} \) has solutions iff \( \gcd(a, m) \mid b \).
- **Proof involves two steps:**
  1. If \( ax \equiv b \pmod{m} \) has solutions, then \( \gcd(a, m) \mid b \).
  2. If \( \gcd(a, m) \mid b \), then \( ax \equiv b \pmod{m} \) has solutions.
- **First prove (1), then (2).**

Proof, Part I

If \( ax \equiv b \pmod{m} \) has solutions, then \( \gcd(a, m) \mid b \).

Proof, Part II

If \( \gcd(a, m) \mid b \), then \( ax \equiv b \pmod{m} \) has solutions.

- Let \( d = \gcd(a, m) \) and suppose \( d \mid b \).
- Then, there is a \( k \) such that \( b = dk \).
- By earlier theorem, there exist \( s, t \) such that \( d = s \cdot a + t \cdot m \).
- Multiply both sides by \( k \): \( dk = a \cdot (sk) + m \cdot (tk) \).
- Since \( b = dk \), we have \( b \equiv a \cdot (sk) \pmod{m} \).
- Thus, \( b \equiv a \cdot (sk) \pmod{m} \).
- Hence, \( sk \) is a solution. \( \Box \)
Examples

- Does $5x \equiv 7 \pmod{15}$ have any solutions?
- Does $3x \equiv 4 \pmod{7}$ have any solutions?

Finding Solutions

- Can determine existence of solutions, but how to find them?
- Theorem: Let $d = \gcd(a, m) = sa + tm$. If $d | b$, then the solutions to $ax \equiv b \pmod{m}$ are given by:
  
  $x = \frac{sb}{d} + \frac{m}{d} u$ where $u \in \mathbb{Z}$

Example

Let $d = \gcd(a, m) = sa + tm$. If $d | b$, then the solutions to $ax \equiv b \pmod{m}$ are given by:

$x = \frac{sb}{d} + \frac{m}{d} u$ where $u \in \mathbb{Z}$

- What are the solutions to the linear congruence $3x \equiv 4 \pmod{7}$?

Another Example

Let $d = \gcd(a, m) = sa + tm$. If $d | b$, then the solutions to $ax \equiv b \pmod{m}$ are given by:

$x = \frac{sb}{d} + \frac{m}{d} u$ where $u \in \mathbb{Z}$

- What are the solutions to the linear congruence $3x \equiv 1 \pmod{7}$?

Inverse Modulo $m$

- The inverse of $a$ modulo $m$, written $\overline{a}$ has the property:
  
  $a\overline{a} \equiv 1 \pmod{m}$

- Theorem: Inverse of $a$ modulo $m$ exists if and only if $a$ and $m$ are relatively prime.

- Does 3 have an inverse modulo 7?

Example

- Find an inverse of 3 modulo 7.

An inverse is any solution to $3x \equiv 1 \pmod{7}$

Earlier, we already computed solutions for this equation as:

$x = -2 + 7u$

Thus, $-2$ is an inverse of 3 modulo 7

5, 12, $-9$, . . . are also inverses
Example 2

- Find inverse of 2 modulo 5.

Cryptography

- Cryptography is the study of techniques for secure transmission of information in the presence of adversaries.

Private vs. Public Crypto Systems

- Two different kinds of cryptography systems:
  1. Private key cryptography (also known as symmetric)
  2. Public key cryptography (asymmetric)

Private Key Cryptography

- Public key cryptography is the modern method: different keys are used to encrypt vs. decrypt message
- Most commonly used public key system is RSA
- Great application of number theory and things we’ve learned

RSA History

- Named after its inventors Rivest, Shamir, and Adleman, all researchers at MIT (1978)
- Actually, similar system invented earlier by British researcher Clifford Cocks, but classified – unknown until 90’s
RSA Overview

- Bob has two keys: public and private
- Everyone knows Bob’s public key, but only he knows his private key
- Alice encrypts message using Bob’s public key
- Bob decrypts message using private key
- Since public key cannot decrypt, no one can read message except Bob

High Level Math Behind RSA

- In the RSA system, private key consists of two very large prime numbers \( p, q \)
- Public key consists of a number \( n \), which is the product of \( p, q \) and another number \( e \), which is relatively prime with \((p - 1)(q - 1)\)
- Encrypt messages using \( n, e \), but to decrypt, must know \( p, q \)
- In theory, can extract \( p, q \) from \( n \) using prime factorization, but this is intractable for very large numbers
- Security of RSA relies on inherent computational difficulty of prime factorization

Encryption in RSA

- To send message to Bob, Alice first represents message as a sequence of numbers
- Call this number representing message \( M \)
- Alice then uses Bob’s public key \( n, e \) to perform encryption as:
  \[ C = M^e \mod n \]
- \( C \) is called the ciphertext

RSA Decryption

- Decryption key \( d \) is the inverse of \( e \) modulo \((p - 1)(q - 1)\):
  \[ d \cdot e \equiv 1 \mod((p - 1)(q - 1)) \]
- Decryption function: \( C^d \mod n \)
- As we saw earlier, \( d \) can be computed reasonably efficiently if we know \((p - 1)(q - 1)\)
- However, since adversaries do not know \( p, q \), they cannot compute \( d \) with reasonable computational effort!

Security of RSA

- The encryption function used in RSA is a trapdoor function
- Trapdoor function is easy to compute in one direction, but very difficult in reverse direction without additional knowledge
- Decryption without private key is very hard because requires prime factorization (which is intractable for large enough numbers)
- Interesting fact: There are efficient (poly-time) prime factorization algorithms for quantum computers (e.g., Shor’s algorithm)
- If we could build quantum computers with sufficient “qubits”, RSA would no longer be secure!