Announcements

- Fourth homework assignment handed out today
- Due October 18 (Thursday after fall break)
- Covers sequences, countability, number theory

Review of Last Lecture

- Congruence Modulo: \( a \equiv b \pmod{m} \) iff \( m \mid (a - b) \)
- Alternatively, \( a \equiv b \pmod{m} \) iff \( a \mod m = b \mod m \)
- \( \gcd(a, b) \) is the largest integer \( d \) such that \( d \mid a \) and \( d \mid b \)
- Theorem: Let \( a = bq + r \). Then, \( \gcd(a, b) = \gcd(b, r) \)
- Euclid’s algorithm is used to efficiently compute \( \gcd \) of two numbers and is based on previous theorem.

Euclidian GCD Algorithm

- Find \( \gcd \) of 72 and 20
  - 12 = 72 \% 20
  - 8 = 20 \% 12
  - 4 = 12 \% 8
  - 0 = 8 \% 4
- \( \gcd \) 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, Euclidian algorithm gives us a way to compute these integers \( s \) and \( t \)

Example

- Express \( \gcd(252, 198) \) as a linear combination of 252 and 198
  - First apply Euclid’s algorithm (write \( a = bq + r \) at each step):
    1. \( 252 = 1 \cdot 198 + 54 \)
    2. \( 198 = 3 \cdot 54 + 36 \)
    3. \( 54 = 1 \cdot 36 + 18 \)
    4. \( 36 = 2 \cdot 18 + 0 \) \( \Rightarrow \) \( \gcd \) is 18
  - Now, using (3), write 18 as 54 – 1 \cdot 36
  - Using (2), write 18 as 54 – 1 \cdot (198 – 3 \cdot 54)
  - Using (1), we have 54 = 252 – 1 \cdot 198, thus:
    \[
    18 = (252 - 1 \cdot 198) - 1(198 - 3 \cdot (252 - 1 \cdot 198))
    \]
Example, cont.

18 = (252 – 1 · 198) – 1(198 – 3 · (252 – 1 · 198))

- Now, let’s simplify this:
  18 = 252 – 1 · 198 – 1 · 198 + 3 · 252 – 3 · 198

- Now, collect all 252 and 198 terms together:
  18 = 4 · 252 – 5 · 198

- Trace steps of Euclid’s algorithm backwards to derive $s, t$:
  \[ \text{gcd}(a, b) = s \cdot a + t \cdot b \]

- This is known as the extended Euclidean algorithm

---

Another Useful Result

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

- **Proof**: Since $ca \equiv cb \pmod{m}$, there exist $s, t$ 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 $c = csa + ctb$

---

Example

- **Lemma**: If $a, b$ are relatively prime and $a|bc$, then $a|c$.

  - **Suppose** $15 \mid 16 \cdot x$
  
  - **Here** 15 and 16 are relatively prime
  
  - **Thus**, previous theorem implies: $15|x$

---

Question

- **Suppose** $ca \equiv cb \pmod{m}$. Does this imply $a \equiv b \pmod{m}$?

  - **Counterexample**: Consider $14 \equiv 8 \pmod{6}$

  - Thus, $2 \cdot 7 \equiv 2 \cdot 4 \pmod{6}$

  - But $7 \not\equiv 4 \pmod{6}$

  - Therefore, this implication does not hold in the general case!

  - However, if $c$ and $m$ are relatively prime, it does hold

---

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 $c = csa + ctb$

---

Examples

- If $15x \equiv 15y \pmod{4}$, is $x \equiv y \pmod{4}$?

  - **Counterexample**: $8 \cdot 2 \equiv 8 \cdot 3 \pmod{4}$, but $2 \not\equiv 3 \pmod{4}$

- 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?
- Observe: Determining if this congruence has a solution is the same as determining if the equality
  \[ ax - mk = b \]
  has integer solutions.

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 \( \gcd(a, m) \mid b \).

- Suppose \( c \) is a solution, i.e., \( ac \equiv b \pmod{m} \)
- Then, \( m(\gcd(a, m)) \)
- Means there is a \( k \) such that \( ac - b = mk \)
- Rewrite as \( b = ac - mk \)
- \( \gcd(a, m) \mid a \) and \( \gcd(a, m) \mid m \); hence, \( \gcd(a, m) \mid (ac - mk) \)
- Since \( b = ac - mk \), we have \( \gcd(a, m) \mid b \)

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 \( b - a \cdot (sk) = m \cdot tk \)
- Thus, \( b \equiv a \cdot (sk) \pmod{m} \)
- Hence, \( sk \) is a solution.

Examples

- Does \( 5x \equiv 7 \pmod{15} \) have any solutions?
- Does \( 3x \equiv 4 \pmod{7} \) have any solutions?
- Note: This result generalizes to linear Diophantine equations
- Equality \( a_1x_1 + a_2x_2 + \ldots + a_nx_n = b \) has integer solutions iff \( \gcd(a_1, a_2, \ldots, a_n) \mid b \)
- Previous result just an instance of this because \( ax \equiv b \pmod{m} \) can be written as \( ax - mk = b \)

Examples

- Does \( 77x + 42y = 35 \) have integer solutions?
- Does \( 6x + 9y + 12z = 7 \) have integer 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:
  \[ x = \frac{sb}{d} + \frac{m}{d} u \quad \text{where} \quad u \in \mathbb{Z} \]

Another Example

Let \( d = \gcd(a, m) = sa + tm \). If \( d | b \), then the solutions to \( ax \equiv b \pmod{m} \) are given by:
\[ x = \frac{sb}{d} + \frac{m}{d} u \quad \text{where} \quad u \in \mathbb{Z} \]

- What are the solutions to the linear congruence \( 3x \equiv 1 \pmod{7} \)?
- First, need to find \( s, t \) such that \( 3s + 7t = \gcd(3, 7) \)
- Apply Euclidean algorithm: \( 7 = 2 \cdot 3 + 1 \) and \( 3 = 3 \cdot 1 + 0 \)
- Hence \( \gcd(3, 7) = 1 = 2 \cdot 3 + 1 \cdot 7 \). Thus, \( s = -2 \) and \( t = 1 \)
- **Solutions:** \( x = -2 \cdot 4 + 7u = -8 + 7u \) (e.g., \(-8, -1, 6, 13, \ldots\))

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, \(-9\), \ldots are also inverses

Example 2

- Find inverse of 2 modulo 5.
- Need to solve the congruence \( 2x \equiv 1 \pmod{5} \)
- What are \( s, t \) such that \( 2s + 5t = 17 \)
- One solution: \( \frac{1}{2} \equiv 3 \)
- Other solutions: 8, 13, 18, \ldots
### Solving Systems of Linear Congruences

- So far, learned how to solve single linear congruence
- In some cases, need to solve a system of linear congruences
- A famous theorem, called **Chinese remainder theorem**, tells us how to solve a system of linear congruences
- **Chinese Remainder Theorem**: Let \( m \) and \( n \) be relatively prime integers. Then, the system:
  
  \[
  x \equiv a \pmod{m} \\
  x \equiv b \pmod{n}
  \]
  
  has a solution. Furthermore, all solutions are congruent to \( ant + bms \) modulo \( mn \) where \( ms + nt = 1 \).

### Example

- Now, combine this with last congruence:
  
  \[
  x \equiv 8 \pmod{15} \\
  x \equiv 2 \pmod{7}
  \]
  
  - Find all solutions to the following system:
    
    \[
    x \equiv 2 \pmod{3} \\
    x \equiv 3 \pmod{5} \\
    x \equiv 2 \pmod{7}
    \]
  
  - Use Euclid’s algorithm: \( 5 = 1 \cdot 3 + 2, 3 = 1 \cdot 2 + 1, 2 = 2 \cdot 1 + 0 \)
  - Applying backward substitution, we get: \( 1 = 2 \cdot 3 - 1 \cdot 5 \)
  - Hence, \( s = 2, t = -1 \) and \( ant + bms = -10 + 18 = 8 \)
  - Thus, solution to first two congruences: \( x \equiv 8 \pmod{105} \)

### Cryptography

- **Cryptography**: the study of techniques for secure transmission of information in the presence of adversaries

  - How can Alice send secret messages to Bob without Eve being able to read them?

### Private vs. Public Crypto Systems

- Two different kinds of cryptography systems:
  1. **Private (secret) key cryptography**
  2. **Public key cryptography**

- In **private key cryptography**, sender and receiver agree on **secret key** that both use to encrypt/decrypt the message

- In **public key cryptography**, a **public key** is used to encrypt the message, and a **private key** is used to decrypt the message

### Private Key Cryptography

- **Private key crypto** is classical method, used since antiquity
- Caesar’s cipher is an example of private key cryptography
- Caesar’s cipher is **shift cipher** where \( f(p) = (p + k) \pmod{26} \)
- Both receiver and sender need to know \( k \) to encrypt/decrypt

  - **Analogy**: Alice wants to send Bob briefcase with secret message; they have a common key to lock/unlock briefcase

- Alice, locks briefcase with shared key and Bob unlocks brief case with shared key

- Only works well when number of parties involved in communicated is small
Public Key Cryptography

- Public key cryptography is the modern method, proposed by Diffie and Hellman in 70’s
- Different keys are used to encrypt vs. decrypt message
- How can parties exchange information using different keys?
- **Analogy**: Alice puts message in briefcase, locks with her own key $A$, sends to Bob
- Bob gets locked briefcase, adds his lock $B$, sends back to Alice
- Alice gets double locked box, removes $A$, sends back to Bob
- Bob opens briefcase using his own key

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

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 $e$
- $e$ is a number relatively prime with $(p−1)(q−1)$ ($\phi(N)$, Euler’s totient function)
- 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:
  \[ C = M^e \pmod{n} \]
- $C$ is called the **ciphertext**
### Encryption Example

- Encrypt message "STOP" using RSA with \( n = 2537 \), \( e = 13 \)
- First convert each letter to a number in \([0, 25]\):
  \( S = 18 \), \( T = 19 \), \( O = 14 \), \( P = 15 \)
- Group sequence into blocks of 4 digits:
  \( M = 1819 \ 1415 \)
- Now encrypt each block as \( C = M^{13} \mod 2537 \)
- For first block, \( 1819^{13} \mod 2537 = 2081 \); for second block \( 1415^{13} \mod 2537 = 2182 \)
- **Ciphertext:** 2081 2182

### RSA Decryption

- How do we decrypt cipher text using private keys \( p \), \( q \)?
- **Decryption key** \( d \) is the inverse of \( e \) modulo \((p - 1)(q - 1)\):
  \[ d \cdot e \equiv 1 \pmod{(p - 1)(q - 1)} \]
- 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!

### Decryption Example

- Decrypt the cipher text 0981 0461 for the RSA cipher with \( p = 43 \), \( q = 59 \), and \( e = 13 \).
- First we need to compute \( s \), \( t \), such that:
  \[ 13s + 2436t = 1 \]
- **Apply extended Euclidean algorithm:** \( s = 937 \), \( t = -5 \)

### Example, cont.

Decrypted 0981 0461 using \( p = 43 \), \( q = 59 \), \( n = 2537 \), and \( e = 13 \).
- To solve \( 13x \equiv 1 \pmod{2436} \), computed \( s = 937 \), \( t = -5 \)
- **Recall:** Solution to this system is given by:
  \[ x = \frac{sb}{d} + \frac{my}{d}u \quad \text{where } u \in \mathbb{Z} \]
- Here, \( s = 937 \), \( b, d = 1 \), \( m = 2436 \), thus solution: \( x = 937 \)
- 0981\(^{937} \mod 2537) = 0704; 0461\(^{937} \mod 2537) = 1115
- Thus, decrypted message is **0704 1115**, or in English, "HELP"

### 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
- Encryption direction is easy because just requires exponentiation and mod
- Decryption without private key is very hard because requires prime factorization
- Therefore, security of RSA depends on difficulty of prime factorization
Security of RSA, cont.

- However, as computers get more powerful and factorization algorithms better, possible to factor larger and larger integers
- Therefore, over time, necessary to use larger and larger prime numbers to ensure secure communication
- For quantum computing, there are very efficient algorithms for computing prime factors (Shor’s algorithm)
- If we could build quantum computers with sufficient “qubits”, RSA would no longer be secure!
- However, today, RSA is considered secure if you use sufficiently large prime numbers (> 200 digits)

Book Recommendation

If you are interested in (history of) cryptography, read “The Code Book” by Simon Singh!