Number Theory and Cryptography

Chapter 4: Part II

UWO – November 6, 2021

Number Theory and Cryptography

Chapter 4: Part II

UWO – November 6, 2021

Plan for Part II

1. Solving Congruences

1.1 Linear Congruences
1.2 Systems of Linear Congruences

2. Applications of Congruences

2.1 Hashing Functions
2.2 Pseudorandom Numbers
2.3 Checking Digits

3. Cryptography

3.1 Classical cryptography
3.2 Public Key Cryptography
3.3 The RSA Encryption

Plan for Part II

Linear congruences

Definition

A congruence of the form $ax \equiv b \pmod{m}$, where m is a positive integer, a and b are integers, and x is a variable, is called a *linear congruence*.

Linear congruences

Definition

A congruence of the form $ax \equiv b \pmod{m}$, where m is a positive integer, a and b are integers, and x is a variable, is called a *linear congruence*. The solutions to a linear congruence $ax \equiv b \pmod{m}$ are all integers x that satisfy the congruence.

Linear congruences

Definition

A congruence of the form $ax \equiv b \pmod{m}$, where m is a positive integer, a and b are integers, and x is a variable, is called a *linear congruence*. The solutions to a linear congruence $ax \equiv b \pmod{m}$ are all integers x that satisfy the congruence.

Definition

An integer $\overline{a}$ such that $\overline{a}a \equiv 1 \pmod{m}$ is said to be an *inverse* of a modulo m.

Linear congruences

Definition

A congruence of the form $ax \equiv b \pmod{m}$, where m is a positive integer, a and b are integers, and x is a variable, is called a *linear congruence*. The solutions to a linear congruence $ax \equiv b \pmod{m}$ are all integers x that satisfy the congruence.

Definition

An integer $\overline{a}$ such that $\overline{a}a \equiv 1 \pmod{m}$ is said to be an *inverse* of a modulo m.

Example

5 is an inverse of 3 modulo 7 since $5 \cdot 3 = 15 \equiv 1 \pmod{7}$

Linear congruences

Definition

A congruence of the form $ax \equiv b \pmod{m}$, where m is a positive integer, a and b are integers, and x is a variable, is called a *linear congruence*. The solutions to a linear congruence $ax \equiv b \pmod{m}$ are all integers x that satisfy the congruence.

Definition

An integer $\overline{a}$ such that $\overline{a}a \equiv 1 \pmod{m}$ is said to be an *inverse* of a modulo m.

Example

5 is an inverse of 3 modulo 7 since $5 \cdot 3 = 15 \equiv 1 \pmod{7}$

One method of solving linear congruences makes use of an inverse $\overline{a}$, if it exists.

Linear congruences

Definition

A congruence of the form $ax \equiv b \pmod{m}$, where m is a positive integer, a and b are integers, and x is a variable, is called a *linear congruence*. The solutions to a linear congruence $ax \equiv b \pmod{m}$ are all integers x that satisfy the congruence.

Definition

An integer $\overline{a}$ such that $\overline{a}a \equiv 1 \pmod{m}$ is said to be an *inverse* of a modulo m.

Example

5 is an inverse of 3 modulo 7 since $5 \cdot 3 = 15 \equiv 1 \pmod{7}$

One method of solving linear congruences makes use of an inverse $\overline{a}$, if it exists. Although we can not divide both sides of the congruence by a, we can multiply by $\overline{a}$ to solve for x.

Linear congruences

Definition

A congruence of the form $ax \equiv b \pmod{m}$, where m is a positive integer, a and b are integers, and x is a variable, is called a *linear congruence*. The solutions to a linear congruence $ax \equiv b \pmod{m}$ are all integers x that satisfy the congruence.

Definition

An integer $\overline{a}$ such that $\overline{a}a \equiv 1 \pmod{m}$ is said to be an *inverse* of a modulo m.

Example

5 is an inverse of 3 modulo 7 since $5 \cdot 3 = 15 \equiv 1 \pmod{7}$

One method of solving linear congruences makes use of an inverse $\overline{a}$, if it exists. Although we can not divide both sides of the congruence by a, we can multiply by $\overline{a}$ to solve for x. Indeed, we have:

$$ax \equiv b \pmod{m} \rightarrow \quad \overline{a}ax \equiv \overline{a}b \pmod{m} \rightarrow \quad x \equiv \overline{a}b \pmod{m}$$

Computing the inverse of a modulo m

The following theorem guarantees that an inverse of a modulo m exists whenever a and m are relatively prime, that is when $\gcd(a, m) = 1$.

Computing the inverse of a modulo m

The following theorem guarantees that an inverse of a modulo m exists whenever a and m are relatively prime, that is when $\gcd(a, m) = 1$.

Theorem

If a and m are relatively prime integers and $m > 1$, then an inverse of a modulo m exists.

Computing the inverse of a modulo m

The following theorem guarantees that an inverse of a modulo m exists whenever a and m are relatively prime, that is when $\gcd(a, m) = 1$.

Theorem

If a and m are relatively prime integers and $m > 1$, then an inverse of a modulo m exists. Furthermore, this inverse is unique modulo m (that is, there is a unique positive integer $\overline{a}$ less than m that is an inverse of a modulo m).

Proof.

Since $\gcd(a, m) = 1$, by Bézout's Theorem, there are integers s and t such that $sa + tm = 1$.

Computing the inverse of a modulo m

The following theorem guarantees that an inverse of a modulo m exists whenever a and m are relatively prime, that is when $\gcd(a, m) = 1$.

Theorem

If a and m are relatively prime integers and $m > 1$, then an inverse of a modulo m exists. Furthermore, this inverse is unique modulo m (that is, there is a unique positive integer $\overline{a}$ less than m that is an inverse of a modulo m).

Proof.

Since $\gcd(a, m) = 1$, by Bézout's Theorem, there are integers s and t such that $sa + tm = 1$.

1. Hence, $tm = 1 - sa$.

Computing the inverse of a modulo m

The following theorem guarantees that an inverse of a modulo m exists whenever a and m are relatively prime, that is when $\gcd(a, m) = 1$.

Theorem

If a and m are relatively prime integers and $m > 1$, then an inverse of a modulo m exists. Furthermore, this inverse is unique modulo m (that is, there is a unique positive integer $\overline{a}$ less than m that is an inverse of a modulo m).

Proof.

Since $\gcd(a, m) = 1$, by Bézout's Theorem, there are integers s and t such that $sa + tm = 1$.

1. Hence, $tm = 1 - sa$.
2. Therefore, m divides $1 - sa$

Computing the inverse of a modulo m

The following theorem guarantees that an inverse of a modulo m exists whenever a and m are relatively prime, that is when $\gcd(a, m) = 1$.

Theorem

If a and m are relatively prime integers and $m > 1$, then an inverse of a modulo m exists. Furthermore, this inverse is unique modulo m (that is, there is a unique positive integer $\overline{a}$ less than m that is an inverse of a modulo m).

Proof.

Since $\gcd(a, m) = 1$, by Bézout's Theorem, there are integers s and t such that $sa + tm = 1$.

1. Hence, $tm = 1 - sa$.
2. Therefore, m divides $1 - sa$
3. According to the definition of congruence, $sa \equiv 1 \pmod{m}$

Computing the inverse of a modulo m

The following theorem guarantees that an inverse of a modulo m exists whenever a and m are relatively prime, that is when $\gcd(a, m) = 1$.

Theorem

If a and m are relatively prime integers and $m > 1$, then an inverse of a modulo m exists. Furthermore, this inverse is unique modulo m (that is, there is a unique positive integer $\overline{a}$ less than m that is an inverse of a modulo m).

Proof.

Since $\gcd(a, m) = 1$, by Bézout's Theorem, there are integers s and t such that $sa + tm = 1$.

1. Hence, $tm = 1 - sa$.
2. Therefore, m divides $1 - sa$
3. According to the definition of congruence, $sa \equiv 1 \pmod{m}$
4. Consequently, s is an inverse of a modulo m.

Computing the inverse of a modulo m

The following theorem guarantees that an inverse of a modulo m exists whenever a and m are relatively prime, that is when $\gcd(a, m) = 1$.

Theorem

If a and m are relatively prime integers and $m > 1$, then an inverse of a modulo m exists. Furthermore, this inverse is unique modulo m (that is, there is a unique positive integer $\overline{a}$ less than m that is an inverse of a modulo m).

Proof.

Since $\gcd(a, m) = 1$, by Bézout's Theorem, there are integers s and t such that $sa + tm = 1$.

1. Hence, $tm = 1 - sa$.
2. Therefore, m divides $1 - sa$
3. According to the definition of congruence, $sa \equiv 1 \pmod{m}$
4. Consequently, s is an inverse of a modulo m.
5. The uniqueness of the inverse is proved in Tutorial 7.

Computing inverses

The (extended) Euclidean algorithm and Bézout coefficients gives us a systematic approaches to finding inverses.

Computing inverses

The (extended) Euclidean algorithm and Bézout coefficients gives us a systematic approaches to finding inverses.

Example

Find an inverse of 3 modulo 7.

Computing inverses

The (extended) Euclidean algorithm and Bézout coefficients gives us a systematic approaches to finding inverses.

Example

Find an inverse of 3 modulo 7.

Solution: Because $\gcd(3, 7) = 1$, an inverse of 3 modulo 7 exists.

Computing inverses

The (extended) Euclidean algorithm and Bézout coefficients gives us a systematic approaches to finding inverses.

Example

Find an inverse of 3 modulo 7.

Solution: Because $\gcd(3,7) = 1$, an inverse of 3 modulo 7 exists.

1. Simply using the division algorithm: $7 = 2 \cdot 3 + 1$.

Computing inverses

The (extended) Euclidean algorithm and Bézout coefficients gives us a systematic approaches to finding inverses.

Example

Find an inverse of 3 modulo 7.

Solution: Because $\gcd(3, 7) = 1$, an inverse of 3 modulo 7 exists.

1. Simply using the division algorithm: $7 = 2 \cdot 3 + 1$.
2. From this equation, we get $-2 \cdot 3 + 1 \cdot 7 = 1$.

Computing inverses

The (extended) Euclidean algorithm and Bézout coefficients gives us a systematic approaches to finding inverses.

Example

Find an inverse of 3 modulo 7.

Solution: Because $\gcd(3,7) = 1$, an inverse of 3 modulo 7 exists.

1. Simply using the division algorithm: $7 = 2 \cdot 3 + 1$.
2. From this equation, we get $-2 \cdot 3 + 1 \cdot 7 = 1$.
3. That is, -2 and 1 are Bézout coefficients of 3 and 7.

Computing inverses

The (extended) Euclidean algorithm and Bézout coefficients gives us a systematic approaches to finding inverses.

Example

Find an inverse of 3 modulo 7.

Solution: Because $\gcd(3, 7) = 1$, an inverse of 3 modulo 7 exists.

1. Simply using the division algorithm: $7 = 2 \cdot 3 + 1$.
2. From this equation, we get $-2 \cdot 3 + 1 \cdot 7 = 1$.
3. That is, -2 and 1 are Bézout coefficients of 3 and 7.
4. Hence, $-2 \cdot 3 \equiv 1 \pmod 7$ and -2 is an inverse of 3 modulo 7.

Computing inverses

The (extended) Euclidean algorithm and Bézout coefficients gives us a systematic approaches to finding inverses.

Example

Find an inverse of 3 modulo 7.

Solution: Because $\gcd(3, 7) = 1$, an inverse of 3 modulo 7 exists.

1. Simply using the division algorithm: $7 = 2 \cdot 3 + 1$.
2. From this equation, we get $-2 \cdot 3 + 1 \cdot 7 = 1$.
3. That is, -2 and 1 are Bézout coefficients of 3 and 7.
4. Hence, $-2 \cdot 3 \equiv 1 \pmod 7$ and -2 is an inverse of 3 modulo 7.
5. Also every integer congruent to -2 modulo 7 is an inverse of 3 modulo 7, i.e., 5, $-9, 12$, etc.

Computing inverses

Find an inverse of 101 modulo 4620.

Computing inverses

Find an inverse of 101 modulo 4620.

1. First use the Euclidean algorithm to show $\gcd(101, 4620) = 1$.

Computing inverses

Find an inverse of 101 modulo 4620.

1. First use the Euclidean algorithm to show $\gcd(101, 4620) = 1$.
2. Second, working backwards to find Bézout coefficients.

1. $4620 = 45 \cdot 101 + 75$

Computing inverses

Find an inverse of 101 modulo 4620.

1. First use the Euclidean algorithm to show $\gcd(101, 4620) = 1$.
2. Second, working backwards to find Bézout coefficients.

1. $4620 = 45 \cdot 101 + 75$
2. $101 = 1 \cdot 75 + 26$

Computing inverses

Find an inverse of 101 modulo 4620.

1. First use the Euclidean algorithm to show $\gcd(101, 4620) = 1$.
2. Second, working backwards to find Bézout coefficients.

1. $4620 = 45 \cdot 101 + 75$
2. $101 = 1 \cdot 75 + 26$
3. $75 = 2 \cdot 26 + 23$

Computing inverses

Find an inverse of 101 modulo 4620.

1. First use the Euclidean algorithm to show $\gcd(101, 4620) = 1$.
2. Second, working backwards to find Bézout coefficients.

1. $4620 = 45 \cdot 101 + 75$
2. $101 = 1 \cdot 75 + 26$
3. $75 = 2 \cdot 26 + 23$
4. $26 = 1 \cdot 23 + 3$

Computing inverses

Find an inverse of 101 modulo 4620.

1. First use the Euclidean algorithm to show $\gcd(101, 4620) = 1$.
2. Second, working backwards to find Bézout coefficients.

1. $4620 = 45 \cdot 101 + 75$
2. $101 = 1 \cdot 75 + 26$
3. $75 = 2 \cdot 26 + 23$
4. $26 = 1 \cdot 23 + 3$
5. $23 = 7 \cdot 3 + 2$

Computing inverses

Find an inverse of 101 modulo 4620.

1. First use the Euclidean algorithm to show $\gcd(101, 4620) = 1$.
2. Second, working backwards to find Bézout coefficients.

1. $4620 = 45 \cdot 101 + 75$
2. $101 = 1 \cdot 75 + 26$
3. $75 = 2 \cdot 26 + 23$
4. $26 = 1 \cdot 23 + 3$
5. $23 = 7 \cdot 3 + 2$
6. $3 = 1 \cdot 2 + 1$

Computing inverses

Find an inverse of 101 modulo 4620.

1. First use the Euclidean algorithm to show $\gcd(101, 4620) = 1$.
2. Second, working backwards to find Bézout coefficients.

1. $4620 = 45 \cdot 101 + 75$
2. $101 = 1 \cdot 75 + 26$
3. $75 = 2 \cdot 26 + 23$
4. $26 = 1 \cdot 23 + 3$
5. $23 = 7 \cdot 3 + 2$
6. $3 = 1 \cdot 2 + 1$
7. $2 = 2 \cdot 1$

Since the last nonzero remainder is 1, $\gcd(101, 4260) = 1$

Computing inverses

Find an inverse of 101 modulo 4620.

1. First use the Euclidean algorithm to show $\gcd(101, 4620) = 1$.
2. Second, working backwards to find Bézout coefficients.

1. $4620 = 45 \cdot 101 + 75$
2. $101 = 1 \cdot 75 + 26$
3. $75 = 2 \cdot 26 + 23$
4. $26 = 1 \cdot 23 + 3$
5. $23 = 7 \cdot 3 + 2$
6. $3 = 1 \cdot 2 + 1$
7. $2 = 2 \cdot 1$

Since the last nonzero remainder is 1, $\gcd(101, 4260) = 1$

8. $1 = 3 - 1 \cdot 2$

Computing inverses

Find an inverse of 101 modulo 4620.

1. First use the Euclidean algorithm to show $\gcd(101, 4620) = 1$.
2. Second, working backwards to find Bézout coefficients.

1. $4620 = 45 \cdot 101 + 75$
2. $101 = 1 \cdot 75 + 26$
3. $75 = 2 \cdot 26 + 23$
4. $26 = 1 \cdot 23 + 3$
5. $23 = 7 \cdot 3 + 2$
6. $3 = 1 \cdot 2 + 1$
7. $2 = 2 \cdot 1$

Since the last nonzero remainder is 1, $\gcd(101, 4260) = 1$

8. $1 = 3 - 1 \cdot 2$
9. $1 = 3 - 1 \cdot (23 - 7 \cdot 3) = -1 \cdot 23 + 8 \cdot 3$

Computing inverses

Find an inverse of 101 modulo 4620.

1. First use the Euclidean algorithm to show $\gcd(101, 4620) = 1$.
2. Second, working backwards to find Bézout coefficients.

1. $4620 = 45 \cdot 101 + 75$
2. $101 = 1 \cdot 75 + 26$
3. $75 = 2 \cdot 26 + 23$
4. $26 = 1 \cdot 23 + 3$
5. $23 = 7 \cdot 3 + 2$
6. $3 = 1 \cdot 2 + 1$
7. $2 = 2 \cdot 1$

Since the last nonzero remainder is 1, $\gcd(101, 4260) = 1$

8. $1 = 3 - 1 \cdot 2$
9. $1 = 3 - 1 \cdot (23 - 7 \cdot 3) = -1 \cdot 23 + 8 \cdot 3$
10. $1 = -1 \cdot 23 + 8 \cdot (26 - 1 \cdot 23) = 8 \cdot 26 - 9 \cdot 23$

Computing inverses

Find an inverse of 101 modulo 4620.

1. First use the Euclidean algorithm to show $\gcd(101, 4620) = 1$.
2. Second, working backwards to find Bézout coefficients.

1. $4620 = 45 \cdot 101 + 75$
2. $101 = 1 \cdot 75 + 26$
3. $75 = 2 \cdot 26 + 23$
4. $26 = 1 \cdot 23 + 3$
5. $23 = 7 \cdot 3 + 2$
6. $3 = 1 \cdot 2 + 1$
7. $2 = 2 \cdot 1$

Since the last nonzero remainder is 1, $\gcd(101, 4260) = 1$

8. $1 = 3 - 1 \cdot 2$
9. $1 = 3 - 1 \cdot (23 - 7 \cdot 3) = -1 \cdot 23 + 8 \cdot 3$
10. $1 = -1 \cdot 23 + 8 \cdot (26 - 1 \cdot 23) = 8 \cdot 26 - 9 \cdot 23$
11. $1 = 8 \cdot 26 - 9 \cdot (75 - 2 \cdot 26) = 26 \cdot 26 - 9 \cdot 75$

Computing inverses

Find an inverse of 101 modulo 4620.

1. First use the Euclidean algorithm to show $\gcd(101, 4620) = 1$.
2. Second, working backwards to find Bézout coefficients.

1. $4620 = 45 \cdot 101 + 75$
2. $101 = 1 \cdot 75 + 26$
3. $75 = 2 \cdot 26 + 23$
4. $26 = 1 \cdot 23 + 3$
5. $23 = 7 \cdot 3 + 2$
6. $3 = 1 \cdot 2 + 1$
7. $2 = 2 \cdot 1$

Since the last nonzero remainder is 1, $\gcd(101, 4260) = 1$

8. $1 = 3 - 1 \cdot 2$
9. $1 = 3 - 1 \cdot (23 - 7 \cdot 3) = -1 \cdot 23 + 8 \cdot 3$
10. $1 = -1 \cdot 23 + 8 \cdot (26 - 1 \cdot 23) = 8 \cdot 26 - 9 \cdot 23$
11. $1 = 8 \cdot 26 - 9 \cdot (75 - 2 \cdot 26) = 26 \cdot 26 - 9 \cdot 75$
12. $1 = 26 \cdot (101 - 1 \cdot 75) - 9 \cdot 75$
    $= 26 \cdot 101 - 35 \cdot 75$

Computing inverses

Find an inverse of 101 modulo 4620.

1. First use the Euclidean algorithm to show $\gcd(101, 4620) = 1$.
2. Second, working backwards to find Bézout coefficients.

1. $4620 = 45 \cdot 101 + 75$
2. $101 = 1 \cdot 75 + 26$
3. $75 = 2 \cdot 26 + 23$
4. $26 = 1 \cdot 23 + 3$
5. $23 = 7 \cdot 3 + 2$
6. $3 = 1 \cdot 2 + 1$
7. $2 = 2 \cdot 1$

Since the last nonzero remainder is 1, $\gcd(101, 4260) = 1$

8. $1 = 3 - 1 \cdot 2$
9. $1 = 3 - 1 \cdot (23 - 7 \cdot 3) = -1 \cdot 23 + 8 \cdot 3$
10. $1 = -1 \cdot 23 + 8 \cdot (26 - 1 \cdot 23) = 8 \cdot 26 - 9 \cdot 23$
11. $1 = 8 \cdot 26 - 9 \cdot (75 - 2 \cdot 26) = 26 \cdot 26 - 9 \cdot 75$
12. $1 = 26 \cdot (101 - 1 \cdot 75) - 9 \cdot 75$
    $= 26 \cdot 101 - 35 \cdot 75$
13. $1 = 26 \cdot 101 - 35 \cdot (4620 - 45 \cdot 101)$
    $= -35 \cdot 4620 + 1601 \cdot 101$

Computing inverses

Find an inverse of 101 modulo 4620.

1. First use the Euclidean algorithm to show $\gcd(101, 4620) = 1$.
2. Second, working backwards to find Bézout coefficients.

1. $4620 = 45 \cdot 101 + 75$
2. $101 = 1 \cdot 75 + 26$
3. $75 = 2 \cdot 26 + 23$
4. $26 = 1 \cdot 23 + 3$
5. $23 = 7 \cdot 3 + 2$
6. $3 = 1 \cdot 2 + 1$
7. $2 = 2 \cdot 1$

Since the last nonzero remainder is 1, $\gcd(101, 4260) = 1$

8. $1 = 3 - 1 \cdot 2$
9. $1 = 3 - 1 \cdot (23 - 7 \cdot 3) = -1 \cdot 23 + 8 \cdot 3$
10. $1 = -1 \cdot 23 + 8 \cdot (26 - 1 \cdot 23) = 8 \cdot 26 - 9 \cdot 23$
11. $1 = 8 \cdot 26 - 9 \cdot (75 - 2 \cdot 26) = 26 \cdot 26 - 9 \cdot 75$
12. $1 = 26 \cdot (101 - 1 \cdot 75) - 9 \cdot 75$
    $= 26 \cdot 101 - 35 \cdot 75$
13. $1 = 26 \cdot 101 - 35 \cdot (4620 - 45 \cdot 101)$
    $= -35 \cdot 4620 + 1601 \cdot 101$

Computing inverses

Find an inverse of 101 modulo 4620.

1. First use the Euclidean algorithm to show $\gcd(101, 4620) = 1$.
2. Second, working backwards to find Bézout coefficients.

1. $4620 = 45 \cdot 101 + 75$
2. $101 = 1 \cdot 75 + 26$
3. $75 = 2 \cdot 26 + 23$
4. $26 = 1 \cdot 23 + 3$
5. $23 = 7 \cdot 3 + 2$
6. $3 = 1 \cdot 2 + 1$
7. $2 = 2 \cdot 1$

Since the last nonzero remainder is 1, $\gcd(101, 4260) = 1$

8. $1 = 3 - 1 \cdot 2$
9. $1 = 3 - 1 \cdot (23 - 7 \cdot 3) = -1 \cdot 23 + 8 \cdot 3$
10. $1 = -1 \cdot 23 + 8 \cdot (26 - 1 \cdot 23) = 8 \cdot 26 - 9 \cdot 23$
11. $1 = 8 \cdot 26 - 9 \cdot (75 - 2 \cdot 26) = 26 \cdot 26 - 9 \cdot 75$
12. $1 = 26 \cdot (101 - 1 \cdot 75) - 9 \cdot 75$
    $= 26 \cdot 101 - 35 \cdot 75$
13. $1 = 26 \cdot 101 - 35 \cdot (4620 - 45 \cdot 101)$
    $= -35 \cdot 4620 + 1601 \cdot 101$
    a. Bézout coefficients for 4620 and 101 are: -35 and 1601

Computing inverses

Find an inverse of 101 modulo 4620.

1. First use the Euclidean algorithm to show $\gcd(101, 4620) = 1$.
2. Second, working backwards to find Bézout coefficients.

1. $4620 = 45 \cdot 101 + 75$
2. $101 = 1 \cdot 75 + 26$
3. $75 = 2 \cdot 26 + 23$
4. $26 = 1 \cdot 23 + 3$
5. $23 = 7 \cdot 3 + 2$
6. $3 = 1 \cdot 2 + 1$
7. $2 = 2 \cdot 1$

Since the last nonzero remainder is 1, $\gcd(101, 4260) = 1$

8. $1 = 3 - 1 \cdot 2$
9. $1 = 3 - 1 \cdot (23 - 7 \cdot 3) = -1 \cdot 23 + 8 \cdot 3$
10. $1 = -1 \cdot 23 + 8 \cdot (26 - 1 \cdot 23) = 8 \cdot 26 - 9 \cdot 23$
11. $1 = 8 \cdot 26 - 9 \cdot (75 - 2 \cdot 26) = 26 \cdot 26 - 9 \cdot 75$
12. $1 = 26 \cdot (101 - 1 \cdot 75) - 9 \cdot 75$
    $= 26 \cdot 101 - 35 \cdot 75$
13. $1 = 26 \cdot 101 - 35 \cdot (4620 - 45 \cdot 101)$
    $= -35 \cdot 4620 + 1601 \cdot 101$
    a. Bézout coefficients for 4620 and 101 are: -35 and 1601
    b. 1601 is an inverse of 101 modulo 4620

Computing inverses

Find an inverse of 101 modulo 4620.

1. First use the Euclidean algorithm to show $\gcd(101, 4620) = 1$.
2. Second, working backwards to find Bézout coefficients.

1. $4620 = 45 \cdot 101 + 75$
2. $101 = 1 \cdot 75 + 26$
3. $75 = 2 \cdot 26 + 23$
4. $26 = 1 \cdot 23 + 3$
5. $23 = 7 \cdot 3 + 2$
6. $3 = 1 \cdot 2 + 1$
7. $2 = 2 \cdot 1$

Since the last nonzero remainder is 1, $\gcd(101, 4260) = 1$

8. $1 = 3 - 1 \cdot 2$
9. $1 = 3 - 1 \cdot (23 - 7 \cdot 3) = -1 \cdot 23 + 8 \cdot 3$
10. $1 = -1 \cdot 23 + 8 \cdot (26 - 1 \cdot 23) = 8 \cdot 26 - 9 \cdot 23$
11. $1 = 8 \cdot 26 - 9 \cdot (75 - 2 \cdot 26) = 26 \cdot 26 - 9 \cdot 75$
12. $1 = 26 \cdot (101 - 1 \cdot 75) - 9 \cdot 75$ $= 26 \cdot 101 - 35 \cdot 75$
13. $1 = 26 \cdot 101 - 35 \cdot (4620 - 45 \cdot 101)$ $= -35 \cdot 4620 + 1601 \cdot 101$
    a. Bézout coefficients for 4620 and 101 are: -35 and 1601
    b. 1601 is an inverse of 101 modulo 4620
    c. Also, -35 is an inverse of 4620 modulo 101

Using inverses to solve congruences

We solve the congruence $ax \equiv b \pmod{m}$ by multiplying both sides by $\bar{a}$.

Using inverses to solve congruences

We solve the congruence $ax \equiv b \pmod{m}$ by multiplying both sides by $\bar{a}$.

Example

What are the solutions of $3x \equiv 4 \pmod{7}$?

Using inverses to solve congruences

We solve the congruence $ax \equiv b \pmod{m}$ by multiplying both sides by $\bar{a}$.

Example

What are the solutions of $3x \equiv 4 \pmod{7}$? **Solution**:

Using inverses to solve congruences

We solve the congruence $ax \equiv b \pmod{m}$ by multiplying both sides by $\bar{a}$.

Example

What are the solutions of $3x \equiv 4 \pmod{7}$? **Solution**:

1. First, $\gcd(3,7) = 1$ and we found that -2 is an inverse of 3 modulo 7 (two slides back).

Using inverses to solve congruences

We solve the congruence $ax \equiv b \pmod{m}$ by multiplying both sides by $\bar{a}$.

Example

What are the solutions of $3x \equiv 4 \pmod{7}$? **Solution**:

1. First, $\gcd(3, 7) = 1$ and we found that -2 is an inverse of 3 modulo 7 (two slides back).
2. We multiply both sides of the congruence by -2 giving $-2 \cdot 3x \equiv -2 \cdot 4 \pmod{7}$.

Using inverses to solve congruences

We solve the congruence $ax \equiv b \pmod{m}$ by multiplying both sides by $\bar{a}$.

Example

What are the solutions of $3x \equiv 4 \pmod{7}$? **Solution**:

1. First, $\gcd(3, 7) = 1$ and we found that -2 is an inverse of 3 modulo 7 (two slides back).
2. We multiply both sides of the congruence by -2 giving $-2 \cdot 3x \equiv -2 \cdot 4 \pmod{7}$.
3. Because $-6 \equiv 1 \pmod{7}$, it follows that if x is a solution then $x \equiv -8 \pmod{7}$ or $x \equiv 6 \pmod{7}$ since $6 \equiv -8 \pmod{7}$

Using inverses to solve congruences

We solve the congruence $ax \equiv b \pmod{m}$ by multiplying both sides by $\bar{a}$.

Example

What are the solutions of $3x \equiv 4 \pmod{7}$? **Solution**:

1. First, $\gcd(3, 7) = 1$ and we found that -2 is an inverse of 3 modulo 7 (two slides back).
2. We multiply both sides of the congruence by -2 giving $-2 \cdot 3x \equiv -2 \cdot 4 \pmod{7}$.
3. Because $-6 \equiv 1 \pmod{7}$, it follows that if x is a solution then $x \equiv -8 \pmod{7}$ or $x \equiv 6 \pmod{7}$ since $6 \equiv -8 \pmod{7}$
4. To verify this solution, assume arbitrary x s.t. $x \equiv 6 \pmod{7}$. It follows that $3x \equiv 3 \cdot 6 \equiv 18 \equiv 4 \pmod{7}$ which shows that all such x satisfy the congruence above.

Using inverses to solve congruences

We solve the congruence $ax \equiv b \pmod{m}$ by multiplying both sides by $\bar{a}$.

Example

What are the solutions of $3x \equiv 4 \pmod{7}$? **Solution**:

1. First, $\gcd(3, 7) = 1$ and we found that -2 is an inverse of 3 modulo 7 (two slides back).
2. We multiply both sides of the congruence by -2 giving $-2 \cdot 3x \equiv -2 \cdot 4 \pmod{7}$.
3. Because $-6 \equiv 1 \pmod{7}$, it follows that if x is a solution then $x \equiv -8 \pmod{7}$ or $x \equiv 6 \pmod{7}$ since $6 \equiv -8 \pmod{7}$
4. To verify this solution, assume arbitrary x s.t. $x \equiv 6 \pmod{7}$. It follows that $3x \equiv 3 \cdot 6 \equiv 18 \equiv 4 \pmod{7}$ which shows that all such x satisfy the congruence above.
5. The solutions are the integers x such that $x \equiv 6 \pmod{7}$, namely, $6, 13, 20 \ldots$ and $-1, -8, -15 \ldots$

Plan for Part II

The Chinese Remainder Theorem

Theorem

Let m and n be two relatively prime integers. Let $s, t \in \mathbb{Z}$ be such that $s\,m + t\,n = 1$.

The Chinese Remainder Theorem

Theorem

Let m and n be two relatively prime integers. Let $s, t \in \mathbb{Z}$ be such that $s\,m + t\,n = 1$. The Chinese Remaindering Theorem *states that for every $a, b \in \mathbb{Z}$ there exists $c \in \mathbb{Z}$ such that*

$$(\forall x \in \mathbb{Z}) \qquad \begin{cases} x \equiv a \mod m \\ x \equiv b \mod n \end{cases} \iff x \equiv c \mod m\,n \tag{1}$$

The Chinese Remainder Theorem

Theorem

Let m and n be two relatively prime integers. Let $s, t \in \mathbb{Z}$ be such that $s\,m + t\,n = 1$. The Chinese Remaindering Theorem *states that for every $a, b \in \mathbb{Z}$ there exists $c \in \mathbb{Z}$ such that*

$$(\forall x \in \mathbb{Z}) \qquad \begin{cases} x \equiv a \mod m \\ x \equiv b \mod n \end{cases} \iff x \equiv c \mod m\,n \tag{1}$$

where a convenient c is given by

$$c \;=\; a + (b-a)\,s\,m = b + (a-b)\,t\,n. \tag{2}$$

The Chinese Remainder Theorem

Proof.

∎

The Chinese Remainder Theorem

Proof.

1. We first check that the above c satisfies both $c \equiv a \mod m$ and $c \equiv b \mod n$.

$\blacksquare$

The Chinese Remainder Theorem

Proof.

1. We first check that the above c satisfies both $c \equiv a \mod m$ and $c \equiv b \mod n$.
   - a. Observe that Relation (2) implies

$$c \equiv a \mod m \qquad \text{and} \qquad c \equiv b \mod n. \tag{3}$$

■

The Chinese Remainder Theorem

Proof.

1. We first check that the above c satisfies both $c \equiv a \mod m$ and $c \equiv b \mod n$.
   a. Observe that Relation (2) implies

$$c \equiv a \mod m \qquad \text{and} \qquad c \equiv b \mod n. \tag{3}$$

2. Assume that $x \equiv c \mod m\,n$ holds. This implies

$$x \equiv c \mod m \qquad \text{and} \qquad x \equiv c \mod n \tag{4}$$

∎

The Chinese Remainder Theorem

Proof.

1. We first check that the above c satisfies both $c \equiv a \mod m$ and $c \equiv b \mod n$.
   a. Observe that Relation (2) implies

$$c \equiv a \mod m \qquad \text{and} \qquad c \equiv b \mod n. \tag{3}$$

2. Assume that $x \equiv c \mod m\, n$ holds. This implies

$$x \equiv c \mod m \qquad \text{and} \qquad x \equiv c \mod n \tag{4}$$

   Thus Relations (3) and (4) lead to

$$x \equiv a \mod m \qquad \text{and} \qquad x \equiv b \mod n \tag{5}$$

$\blacksquare$

The Chinese Remainder Theorem

Proof.

1. We first check that the above c satisfies both $c \equiv a \mod m$ and $c \equiv b \mod n$.
   a. Observe that Relation (2) implies
   $$c \equiv a \mod m \qquad \text{and} \qquad c \equiv b \mod n. \tag{3}$$
2. Assume that $x \equiv c \mod m\,n$ holds. This implies
   $$x \equiv c \mod m \qquad \text{and} \qquad x \equiv c \mod n \tag{4}$$
   Thus Relations (3) and (4) lead to
   $$x \equiv a \mod m \qquad \text{and} \qquad x \equiv b \mod n \tag{5}$$
3. Conversely
   - $x \equiv a \mod m$ implies $x \equiv c \mod m$ that is m divides $x - c$ and
   - $x \equiv b \mod n$ implies $x \equiv c \mod n$ that is n divides $x - c$.

∎

The Chinese Remainder Theorem

Proof.

1. We first check that the above c satisfies both $c \equiv a \mod m$ and $c \equiv b \mod n$.
   a. Observe that Relation (2) implies

$$c \equiv a \mod m \qquad \text{and} \qquad c \equiv b \mod n. \tag{3}$$

2. Assume that $x \equiv c \mod m\,n$ holds. This implies

$$x \equiv c \mod m \qquad \text{and} \qquad x \equiv c \mod n \tag{4}$$

   Thus Relations (3) and (4) lead to

$$x \equiv a \mod m \qquad \text{and} \qquad x \equiv b \mod n \tag{5}$$

3. Conversely
   - $x \equiv a \mod m$ implies $x \equiv c \mod m$ that is m divides $x - c$ and
   - $x \equiv b \mod n$ implies $x \equiv c \mod n$ that is n divides $x - c$.

   Since m and n are relatively prime it follows that $m\,n$ divides $x - c$.

∎

The Chinese Remainder Theorem

Find all integers x such that $0 \leq x < 15$, $x \equiv 1 \mod 3$ and $x \equiv 2 \mod 5$.

The Chinese Remainder Theorem

Find all integers x such that $0 \leq x < 15$, $x \equiv 1 \mod 3$ and $x \equiv 2 \mod 5$.

1. We apply the Chinese Remainder Theorem (as stated above).

The Chinese Remainder Theorem

Find all integers x such that $0 \leq x < 15$, $x \equiv 1 \mod 3$ and $x \equiv 2 \mod 5$.

1. We apply the Chinese Remainder Theorem (as stated above).
2. Using the notations of the theorem, we have $m = 3$, $n = 5$, $a = 1$, $b = 2$.

The Chinese Remainder Theorem

Find all integers x such that $0 \leq x < 15$, $x \equiv 1 \mod 3$ and $x \equiv 2 \mod 5$.

1. We apply the Chinese Remainder Theorem (as stated above).
2. Using the notations of the theorem, we have $m = 3$, $n = 5$, $a = 1$, $b = 2$.
3. We need s and t such that $s\,m + t\,n = 1$, hence

The Chinese Remainder Theorem

Find all integers x such that $0 \leq x < 15$, $x \equiv 1 \mod 3$ and $x \equiv 2 \mod 5$.

1. We apply the Chinese Remainder Theorem (as stated above).
2. Using the notations of the theorem, we have $m = 3$, $n = 5$, $a = 1$, $b = 2$.
3. We need s and t such that $s\,m + t\,n = 1$, hence
4. we can choose $s = 2$ and $t = -1$.

The Chinese Remainder Theorem

Find all integers x such that $0 \leq x < 15$, $x \equiv 1 \mod 3$ and $x \equiv 2 \mod 5$.

1. We apply the Chinese Remainder Theorem (as stated above).
2. Using the notations of the theorem, we have $m = 3$, $n = 5$, $a = 1$, $b = 2$.
3. We need s and t such that $s\,m + t\,n = 1$, hence
4. we can choose $s = 2$ and $t = -1$.
5. Then, we have
$$c \equiv a + (b - a)\,s\,m \equiv 1 + (2 - 1) \times 2 \times 3 \equiv 7 \mod 15.$$

Plan for Part II

Plan for Part II

Hashing functions

A *hashing function* h assigns memory location $h(k)$ to the record that has k as its key.

Hashing functions

A *hashing function* h assigns memory location $h(k)$ to the record that has k as its key.

1. A common hashing function is $h(k) = k \mod m$, where m is the number of memory locations.

Hashing functions

A *hashing function* h assigns memory location $h(k)$ to the record that has k as its key.

1. A common hashing function is $h(k) = k \mod m$, where m is the number of memory locations.
2. Because h is onto, all memory locations are possible.

Hashing functions

A *hashing function* h assigns memory location $h(k)$ to the record that has k as its key.

1. A common hashing function is $h(k) = k \mod m$, where m is the number of memory locations.
2. Because h is onto, all memory locations are possible.

Let $h(k) = k \mod 111$. This hashing function assigns the records of customers with social security numbers as keys to memory locations in the following manner:

Hashing functions

A *hashing function* h assigns memory location $h(k)$ to the record that has k as its key.

1. A common hashing function is $h(k) = k \mod m$, where m is the number of memory locations.
2. Because h is onto, all memory locations are possible.

Let $h(k) = k \mod 111$. This hashing function assigns the records of customers with social security numbers as keys to memory locations in the following manner:

$$\mathrm{h}(064212848) = 064212848 \mod 111 = 14$$

Hashing functions

A *hashing function* h assigns memory location $h(k)$ to the record that has k as its key.

1. A common hashing function is $h(k) = k \mod m$, where m is the number of memory locations.
2. Because h is onto, all memory locations are possible.

Let $h(k) = k \mod 111$. This hashing function assigns the records of customers with social security numbers as keys to memory locations in the following manner:

$$\mathrm{h}(064212848) = 064212848 \mod 111 = 14$$

Hashing functions

A *hashing function* h assigns memory location $h(k)$ to the record that has k as its key.

1. A common hashing function is $h(k) = k \mod m$, where m is the number of memory locations.
2. Because h is onto, all memory locations are possible.

Let $h(k) = k \mod 111$. This hashing function assigns the records of customers with social security numbers as keys to memory locations in the following manner:

> $\mathrm{h}(064212848) = 064212848 \mod 111 = 14$
> $\mathrm{h}(107405723) = 107405723 \mod 111 = 14$, but since location 14 is already occupied, the record is assigned to the next available position, which is 15.

Hashing functions

A *hashing function* h assigns memory location $h(k)$ to the record that has k as its key.

1. A common hashing function is $h(k) = k \mod m$, where m is the number of memory locations.
2. Because h is onto, all memory locations are possible.

Let $h(k) = k \mod 111$. This hashing function assigns the records of customers with social security numbers as keys to memory locations in the following manner:

h(064212848) = 064212848 mod 111 = 14
h(107405723) = 107405723 mod 111 = 14, but since location 14 is already occupied, the record is assigned to the next available position, which is 15.

1. The hashing function is not one-to-one as there are many more possible keys than memory locations. When more than one record is assigned to the same location, we have a *collision* (resolved by assigning the record to the first free location).

Hashing functions

A *hashing function* h assigns memory location $h(k)$ to the record that has k as its key.

1. A common hashing function is $h(k) = k \mod m$, where m is the number of memory locations.
2. Because h is onto, all memory locations are possible.

Let $h(k) = k \mod 111$. This hashing function assigns the records of customers with social security numbers as keys to memory locations in the following manner:

$\text{h}(064212848) = 064212848 \mod 111 = 14$
$\text{h}(107405723) = 107405723 \mod 111 = 14$, but since location 14 is already occupied, the record is assigned to the next available position, which is 15.

1. The hashing function is not one-to-one as there are many more possible keys than memory locations. When more than one record is assigned to the same location, we have a *collision* (resolved by assigning the record to the first free location).
2. For collision resolution, we can use a *linear probing function*: $h(k,i) = (h(k) + i) \pmod m$, where i runs from 0 to $m - 1$.

Hashing functions

A *hashing function* h assigns memory location $h(k)$ to the record that has k as its key.

1. A common hashing function is $h(k) = k \mod m$, where m is the number of memory locations.
2. Because h is onto, all memory locations are possible.

Let $h(k) = k \mod 111$. This hashing function assigns the records of customers with social security numbers as keys to memory locations in the following manner:

> $h(064212848) = 064212848 \mod 111 = 14$
> $h(107405723) = 107405723 \mod 111 = 14$, but since location 14 is already occupied, the record is assigned to the next available position, which is 15.

1. The hashing function is not one-to-one as there are many more possible keys than memory locations. When more than one record is assigned to the same location, we have a *collision* (resolved by assigning the record to the first free location).
2. For collision resolution, we can use a *linear probing function*: $h(k, i) = (h(k) + i) \pmod{m}$, where i runs from 0 to $m - 1$.
3. There are many other methods of handling with collisions.

Plan for Part II

Pseudorandom numbers

1. Randomly chosen numbers are needed for many purposes, including computer simulations.

Pseudorandom numbers

1. Randomly chosen numbers are needed for many purposes, including computer simulations.
2. *Pseudorandom numbers* are not truly random since they are generated by systematic methods.

Pseudorandom numbers

1. Randomly chosen numbers are needed for many purposes, including computer simulations.
2. *Pseudorandom numbers* are not truly random since they are generated by systematic methods.
3. The *linear congruential method* is one commonly used procedure for generating pseudorandom numbers.

Pseudorandom numbers

1. Randomly chosen numbers are needed for many purposes, including computer simulations.
2. *Pseudorandom numbers* are not truly random since they are generated by systematic methods.
3. The *linear congruential method* is one commonly used procedure for generating pseudorandom numbers.
4. Four integers are needed: the *modulus* m, the *multiplier* a, the *increment* c, and *seed* x_0, with $2 \leq a < m, 0 \leq c < m, 0 \leq x_0 < m$.

Pseudorandom numbers

1. Randomly chosen numbers are needed for many purposes, including computer simulations.
2. *Pseudorandom numbers* are not truly random since they are generated by systematic methods.
3. The *linear congruential method* is one commonly used procedure for generating pseudorandom numbers.
4. Four integers are needed: the *modulus* m, the *multiplier* a, the *increment* c, and *seed* x_0, with $2 \leq a < m, 0 \leq c < m, 0 \leq x_0 < m$.
5. We generate a sequence of pseudorandom numbers $\{x_n\}$ with $0 \leq x_n < m$ for all n, by successively using the recursive function

$$x_{n+1} = (ax_n + c) \pmod{m}$$

Pseudorandom numbers

Compute the terms of the sequence by successively using the congruence $x_{n+1} = (7x_n + 4) \pmod 9$ with $x_0 = 3$.

Pseudorandom numbers

Compute the terms of the sequence by successively using the congruence $x_{n+1} = (7x_n + 4) \pmod 9$ with $x_0 = 3$.

$$x_1 = 7x_0 + 4 \pmod 9 = 7 \cdot 3 + 4 \pmod 9 = 25 \pmod 9 = 7,$$

Pseudorandom numbers

Compute the terms of the sequence by successively using the congruence $x_{n+1} = (7x_n + 4) \pmod 9$ with $x_0 = 3$.

$$
\begin{aligned}
x_1 &= 7x_0 + 4 \pmod 9 &&= 7 \cdot 3 + 4 \pmod 9 &&= 25 \pmod 9 &&= 7, \\
x_2 &= 7x_1 + 4 \pmod 9 &&= 7 \cdot 7 + 4 \pmod 9 &&= 53 \pmod 9 &&= 8,
\end{aligned}
$$

Pseudorandom numbers

Compute the terms of the sequence by successively using the congruence $x_{n+1} = (7x_n + 4) \pmod{9}$ with $x_0 = 3$.

$$
\begin{aligned}
x_1 &= 7x_0 + 4 \pmod{9} &&= 7 \cdot 3 + 4 \pmod{9} &&= 25 \pmod{9} &&= 7,\\
x_2 &= 7x_1 + 4 \pmod{9} &&= 7 \cdot 7 + 4 \pmod{9} &&= 53 \pmod{9} &&= 8,\\
x_3 &= 7x_2 + 4 \pmod{9} &&= 7 \cdot 8 + 4 \pmod{9} &&= 60 \pmod{9} &&= 6,
\end{aligned}
$$

Pseudorandom numbers

Compute the terms of the sequence by successively using the congruence $x_{n+1} = (7x_n + 4) \pmod 9$ with $x_0 = 3$.

$$
\begin{aligned}
x_1 &= 7x_0 + 4 \pmod 9 &&= 7 \cdot 3 + 4 \pmod 9 &&= 25 \pmod 9 &&= 7,\\
x_2 &= 7x_1 + 4 \pmod 9 &&= 7 \cdot 7 + 4 \pmod 9 &&= 53 \pmod 9 &&= 8,\\
x_3 &= 7x_2 + 4 \pmod 9 &&= 7 \cdot 8 + 4 \pmod 9 &&= 60 \pmod 9 &&= 6,\\
x_4 &= 7x_3 + 4 \pmod 9 &&= 7 \cdot 6 + 4 \pmod 9 &&= 46 \pmod 9 &&= 1,
\end{aligned}
$$

Pseudorandom numbers

Compute the terms of the sequence by successively using the congruence $x_{n+1} = (7x_n + 4) \pmod 9$ with $x_0 = 3$.

$$
\begin{aligned}
x_1 &= 7x_0 + 4 \pmod 9 &&= 7 \cdot 3 + 4 \pmod 9 &&= 25 \pmod 9 &&= 7,\\
x_2 &= 7x_1 + 4 \pmod 9 &&= 7 \cdot 7 + 4 \pmod 9 &&= 53 \pmod 9 &&= 8,\\
x_3 &= 7x_2 + 4 \pmod 9 &&= 7 \cdot 8 + 4 \pmod 9 &&= 60 \pmod 9 &&= 6,\\
x_4 &= 7x_3 + 4 \pmod 9 &&= 7 \cdot 6 + 4 \pmod 9 &&= 46 \pmod 9 &&= 1,\\
x_5 &= 7x_4 + 4 \pmod 9 &&= 7 \cdot 1 + 4 \pmod 9 &&= 11 \pmod 9 &&= 2,
\end{aligned}
$$

Pseudorandom numbers

Compute the terms of the sequence by successively using the congruence $x_{n+1} = (7x_n + 4) \pmod 9$ with $x_0 = 3$.

$$
\begin{aligned}
x_1 &= 7x_0 + 4 \pmod 9 &&= 7 \cdot 3 + 4 \pmod 9 &&= 25 \pmod 9 &&= 7,\\
x_2 &= 7x_1 + 4 \pmod 9 &&= 7 \cdot 7 + 4 \pmod 9 &&= 53 \pmod 9 &&= 8,\\
x_3 &= 7x_2 + 4 \pmod 9 &&= 7 \cdot 8 + 4 \pmod 9 &&= 60 \pmod 9 &&= 6,\\
x_4 &= 7x_3 + 4 \pmod 9 &&= 7 \cdot 6 + 4 \pmod 9 &&= 46 \pmod 9 &&= 1,\\
x_5 &= 7x_4 + 4 \pmod 9 &&= 7 \cdot 1 + 4 \pmod 9 &&= 11 \pmod 9 &&= 2,\\
x_6 &= 7x_5 + 4 \pmod 9 &&= 7 \cdot 2 + 4 \pmod 9 &&= 18 \pmod 9 &&= 0,
\end{aligned}
$$

Pseudorandom numbers

Compute the terms of the sequence by successively using the congruence $x_{n+1} = (7x_n + 4) \pmod 9$ with $x_0 = 3$.

$$
\begin{aligned}
x_1 &= 7x_0 + 4 \pmod 9 &&= 7 \cdot 3 + 4 \pmod 9 &&= 25 \pmod 9 &&= 7, \\
x_2 &= 7x_1 + 4 \pmod 9 &&= 7 \cdot 7 + 4 \pmod 9 &&= 53 \pmod 9 &&= 8, \\
x_3 &= 7x_2 + 4 \pmod 9 &&= 7 \cdot 8 + 4 \pmod 9 &&= 60 \pmod 9 &&= 6, \\
x_4 &= 7x_3 + 4 \pmod 9 &&= 7 \cdot 6 + 4 \pmod 9 &&= 46 \pmod 9 &&= 1, \\
x_5 &= 7x_4 + 4 \pmod 9 &&= 7 \cdot 1 + 4 \pmod 9 &&= 11 \pmod 9 &&= 2, \\
x_6 &= 7x_5 + 4 \pmod 9 &&= 7 \cdot 2 + 4 \pmod 9 &&= 18 \pmod 9 &&= 0, \\
x_7 &= 7x_6 + 4 \pmod 9 &&= 7 \cdot 0 + 4 \pmod 9 &&= 4 \pmod 9 &&= 4,
\end{aligned}
$$

Pseudorandom numbers

Compute the terms of the sequence by successively using the congruence $x_{n+1} = (7x_n + 4) \pmod 9$ with $x_0 = 3$.

$$
\begin{aligned}
x_1 &= 7x_0 + 4 \pmod 9 &&= 7 \cdot 3 + 4 \pmod 9 &&= 25 \pmod 9 &&= 7,\\
x_2 &= 7x_1 + 4 \pmod 9 &&= 7 \cdot 7 + 4 \pmod 9 &&= 53 \pmod 9 &&= 8,\\
x_3 &= 7x_2 + 4 \pmod 9 &&= 7 \cdot 8 + 4 \pmod 9 &&= 60 \pmod 9 &&= 6,\\
x_4 &= 7x_3 + 4 \pmod 9 &&= 7 \cdot 6 + 4 \pmod 9 &&= 46 \pmod 9 &&= 1,\\
x_5 &= 7x_4 + 4 \pmod 9 &&= 7 \cdot 1 + 4 \pmod 9 &&= 11 \pmod 9 &&= 2,\\
x_6 &= 7x_5 + 4 \pmod 9 &&= 7 \cdot 2 + 4 \pmod 9 &&= 18 \pmod 9 &&= 0,\\
x_7 &= 7x_6 + 4 \pmod 9 &&= 7 \cdot 0 + 4 \pmod 9 &&= 4 \pmod 9 &&= 4,\\
x_8 &= 7x_7 + 4 \pmod 9 &&= 7 \cdot 4 + 4 \pmod 9 &&= 32 \pmod 9 &&= 5,
\end{aligned}
$$

Pseudorandom numbers

Compute the terms of the sequence by successively using the congruence $x_{n+1} = (7x_n + 4) \pmod 9$ with $x_0 = 3$.

$$
\begin{aligned}
x_1 &= 7x_0 + 4 \pmod 9 &&= 7 \cdot 3 + 4 \pmod 9 &&= 25 \pmod 9 &&= 7,\\
x_2 &= 7x_1 + 4 \pmod 9 &&= 7 \cdot 7 + 4 \pmod 9 &&= 53 \pmod 9 &&= 8,\\
x_3 &= 7x_2 + 4 \pmod 9 &&= 7 \cdot 8 + 4 \pmod 9 &&= 60 \pmod 9 &&= 6,\\
x_4 &= 7x_3 + 4 \pmod 9 &&= 7 \cdot 6 + 4 \pmod 9 &&= 46 \pmod 9 &&= 1,\\
x_5 &= 7x_4 + 4 \pmod 9 &&= 7 \cdot 1 + 4 \pmod 9 &&= 11 \pmod 9 &&= 2,\\
x_6 &= 7x_5 + 4 \pmod 9 &&= 7 \cdot 2 + 4 \pmod 9 &&= 18 \pmod 9 &&= 0,\\
x_7 &= 7x_6 + 4 \pmod 9 &&= 7 \cdot 0 + 4 \pmod 9 &&= 4 \pmod 9 &&= 4,\\
x_8 &= 7x_7 + 4 \pmod 9 &&= 7 \cdot 4 + 4 \pmod 9 &&= 32 \pmod 9 &&= 5,\\
x_9 &= 7x_8 + 4 \pmod 9 &&= 7 \cdot 5 + 4 \pmod 9 &&= 39 \pmod 9 &&= 3
\end{aligned}
$$

Pseudorandom numbers

Compute the terms of the sequence by successively using the congruence $x_{n+1} = (7x_n + 4) \pmod 9$ with $x_0 = 3$.

$$\begin{aligned}
x_1 &= 7x_0 + 4 \pmod 9 &&= 7 \cdot 3 + 4 \pmod 9 &&= 25 \pmod 9 &&= 7,\\
x_2 &= 7x_1 + 4 \pmod 9 &&= 7 \cdot 7 + 4 \pmod 9 &&= 53 \pmod 9 &&= 8,\\
x_3 &= 7x_2 + 4 \pmod 9 &&= 7 \cdot 8 + 4 \pmod 9 &&= 60 \pmod 9 &&= 6,\\
x_4 &= 7x_3 + 4 \pmod 9 &&= 7 \cdot 6 + 4 \pmod 9 &&= 46 \pmod 9 &&= 1,\\
x_5 &= 7x_4 + 4 \pmod 9 &&= 7 \cdot 1 + 4 \pmod 9 &&= 11 \pmod 9 &&= 2,\\
x_6 &= 7x_5 + 4 \pmod 9 &&= 7 \cdot 2 + 4 \pmod 9 &&= 18 \pmod 9 &&= 0,\\
x_7 &= 7x_6 + 4 \pmod 9 &&= 7 \cdot 0 + 4 \pmod 9 &&= 4 \pmod 9 &&= 4,\\
x_8 &= 7x_7 + 4 \pmod 9 &&= 7 \cdot 4 + 4 \pmod 9 &&= 32 \pmod 9 &&= 5,\\
x_9 &= 7x_8 + 4 \pmod 9 &&= 7 \cdot 5 + 4 \pmod 9 &&= 39 \pmod 9 &&= 3
\end{aligned}$$

1. The sequence generated is $3, 7, 8, 6, 1, 2, 0, 4, 5, 3, 7, 8, 6, 1, 2, 0, 4, 5, 3, \ldots$

Pseudorandom numbers

Compute the terms of the sequence by successively using the congruence $x_{n+1} = (7x_n + 4) \pmod 9$ with $x_0 = 3$.

$$
\begin{aligned}
x_1 &= 7x_0 + 4 \pmod 9 &&= 7 \cdot 3 + 4 \pmod 9 &&= 25 \pmod 9 &&= 7,\\
x_2 &= 7x_1 + 4 \pmod 9 &&= 7 \cdot 7 + 4 \pmod 9 &&= 53 \pmod 9 &&= 8,\\
x_3 &= 7x_2 + 4 \pmod 9 &&= 7 \cdot 8 + 4 \pmod 9 &&= 60 \pmod 9 &&= 6,\\
x_4 &= 7x_3 + 4 \pmod 9 &&= 7 \cdot 6 + 4 \pmod 9 &&= 46 \pmod 9 &&= 1,\\
x_5 &= 7x_4 + 4 \pmod 9 &&= 7 \cdot 1 + 4 \pmod 9 &&= 11 \pmod 9 &&= 2,\\
x_6 &= 7x_5 + 4 \pmod 9 &&= 7 \cdot 2 + 4 \pmod 9 &&= 18 \pmod 9 &&= 0,\\
x_7 &= 7x_6 + 4 \pmod 9 &&= 7 \cdot 0 + 4 \pmod 9 &&= 4 \pmod 9 &&= 4,\\
x_8 &= 7x_7 + 4 \pmod 9 &&= 7 \cdot 4 + 4 \pmod 9 &&= 32 \pmod 9 &&= 5,\\
x_9 &= 7x_8 + 4 \pmod 9 &&= 7 \cdot 5 + 4 \pmod 9 &&= 39 \pmod 9 &&= 3
\end{aligned}
$$

1. The sequence generated is $3, 7, 8, 6, 1, 2, 0, 4, 5, 3, 7, 8, 6, 1, 2, 0, 4, 5, 3, \ldots$
2. It repeats after generating 9 terms.

Pseudorandom numbers

Compute the terms of the sequence by successively using the congruence $x_{n+1} = (7x_n + 4) \pmod 9$ with $x_0 = 3$.

$$\begin{aligned}
x_1 &= 7x_0 + 4 \pmod 9 &&= 7\cdot 3 + 4 \pmod 9 &&= 25 \pmod 9 &&= 7,\\
x_2 &= 7x_1 + 4 \pmod 9 &&= 7\cdot 7 + 4 \pmod 9 &&= 53 \pmod 9 &&= 8,\\
x_3 &= 7x_2 + 4 \pmod 9 &&= 7\cdot 8 + 4 \pmod 9 &&= 60 \pmod 9 &&= 6,\\
x_4 &= 7x_3 + 4 \pmod 9 &&= 7\cdot 6 + 4 \pmod 9 &&= 46 \pmod 9 &&= 1,\\
x_5 &= 7x_4 + 4 \pmod 9 &&= 7\cdot 1 + 4 \pmod 9 &&= 11 \pmod 9 &&= 2,\\
x_6 &= 7x_5 + 4 \pmod 9 &&= 7\cdot 2 + 4 \pmod 9 &&= 18 \pmod 9 &&= 0,\\
x_7 &= 7x_6 + 4 \pmod 9 &&= 7\cdot 0 + 4 \pmod 9 &&= 4 \pmod 9 &&= 4,\\
x_8 &= 7x_7 + 4 \pmod 9 &&= 7\cdot 4 + 4 \pmod 9 &&= 32 \pmod 9 &&= 5,\\
x_9 &= 7x_8 + 4 \pmod 9 &&= 7\cdot 5 + 4 \pmod 9 &&= 39 \pmod 9 &&= 3
\end{aligned}$$

1. The sequence generated is $3, 7, 8, 6, 1, 2, 0, 4, 5, 3, 7, 8, 6, 1, 2, 0, 4, 5, 3, \ldots$
2. It repeats after generating 9 terms.
3. Commonly, computers use a linear congruential generator with increment $c = 0$. This is called a *pure multiplicative generator.*

Pseudorandom numbers

Compute the terms of the sequence by successively using the congruence $x_{n+1} = (7x_n + 4) \pmod 9$ with $x_0 = 3$.

$$
\begin{aligned}
x_1 &= 7x_0 + 4 \pmod 9 && = 7 \cdot 3 + 4 \pmod 9 && = 25 \pmod 9 && = 7, \\
x_2 &= 7x_1 + 4 \pmod 9 && = 7 \cdot 7 + 4 \pmod 9 && = 53 \pmod 9 && = 8, \\
x_3 &= 7x_2 + 4 \pmod 9 && = 7 \cdot 8 + 4 \pmod 9 && = 60 \pmod 9 && = 6, \\
x_4 &= 7x_3 + 4 \pmod 9 && = 7 \cdot 6 + 4 \pmod 9 && = 46 \pmod 9 && = 1, \\
x_5 &= 7x_4 + 4 \pmod 9 && = 7 \cdot 1 + 4 \pmod 9 && = 11 \pmod 9 && = 2, \\
x_6 &= 7x_5 + 4 \pmod 9 && = 7 \cdot 2 + 4 \pmod 9 && = 18 \pmod 9 && = 0, \\
x_7 &= 7x_6 + 4 \pmod 9 && = 7 \cdot 0 + 4 \pmod 9 && = 4 \pmod 9 && = 4, \\
x_8 &= 7x_7 + 4 \pmod 9 && = 7 \cdot 4 + 4 \pmod 9 && = 32 \pmod 9 && = 5, \\
x_9 &= 7x_8 + 4 \pmod 9 && = 7 \cdot 5 + 4 \pmod 9 && = 39 \pmod 9 && = 3
\end{aligned}
$$

1. The sequence generated is $3, 7, 8, 6, 1, 2, 0, 4, 5, 3, 7, 8, 6, 1, 2, 0, 4, 5, 3, \ldots$
2. It repeats after generating 9 terms.
3. Commonly, computers use a linear congruential generator with increment $c = 0$. This is called a *pure multiplicative generator*.
4. Such a generator with modulus $2^{31} - 1$ and multiplier $7^5 = 16,807$ generates $2^{31} - 2$ numbers before repeating.

Plan for Part II

Checking digits: UPCs

1. A common method for detecting errors in strings of digits is to add an extra digit at the end, which is evaluated using a function.

Checking digits: UPCs

1. A common method for detecting errors in strings of digits is to add an extra digit at the end, which is evaluated using a function.
2. If the final digit is not correct, then the string is assumed not to be correct.

Checking digits: UPCs

1. A common method for detecting errors in strings of digits is to add an extra digit at the end, which is evaluated using a function.
2. If the final digit is not correct, then the string is assumed not to be correct.
3. Retail products are identified by their *Universal Product Codes* (*UPC*s). Usually these have 12 decimal digits, the last one being the check digit. The check digit x_{12} is determined by:
   $3x_1 + x_2 + 3x_3 + x_4 + 3x_5 + x_6 + 3x_7 + x_8 + 3x_9 + x_{10} + 3x_{11} + x_{12} \equiv 0 \mod 10$

Checking digits: UPCs

1. A common method for detecting errors in strings of digits is to add an extra digit at the end, which is evaluated using a function.
2. If the final digit is not correct, then the string is assumed not to be correct.
3. Retail products are identified by their *Universal Product Codes* (*UPC*s). Usually these have 12 decimal digits, the last one being the check digit. The check digit x_{12} is determined by:
   $3x_1 + x_2 + 3x_3 + x_4 + 3x_5 + x_6 + 3x_7 + x_8 + 3x_9 + x_{10} + 3x_{11} + x_{12} \equiv 0 \mod 10$

   a. Suppose that the first 11 digits of the UPC are 79357343104. What is the check digit?

Checking digits: UPCs

1. A common method for detecting errors in strings of digits is to add an extra digit at the end, which is evaluated using a function.
2. If the final digit is not correct, then the string is assumed not to be correct.
3. Retail products are identified by their *Universal Product Codes* (*UPC*s). Usually these have 12 decimal digits, the last one being the check digit. The check digit x_{12} is determined by:
   $3x_1 + x_2 + 3x_3 + x_4 + 3x_5 + x_6 + 3x_7 + x_8 + 3x_9 + x_{10} + 3x_{11} + x_{12} \equiv 0 \mod 10$

   a. Suppose that the first 11 digits of the UPC are 79357343104. What is the check digit?
      1. $3 \cdot 7 + 9 + 3 \cdot 3 + 5 + 3 \cdot 7 + 3 + 3 \cdot 4 + 3 + 3 \cdot 1 + 0 + 3 \cdot 4 + x_{12} \equiv 0 \pmod{10}$

Checking digits: UPCs

1. A common method for detecting errors in strings of digits is to add an extra digit at the end, which is evaluated using a function.
2. If the final digit is not correct, then the string is assumed not to be correct.
3. Retail products are identified by their *Universal Product Codes* (*UPC*s). Usually these have 12 decimal digits, the last one being the check digit. The check digit x_{12} is determined by:
   $3x_1 + x_2 + 3x_3 + x_4 + 3x_5 + x_6 + 3x_7 + x_8 + 3x_9 + x_{10} + 3x_{11} + x_{12} \equiv 0 \mod 10$

   a. Suppose that the first 11 digits of the UPC are 79357343104. What is the check digit?
      1. $3 \cdot 7 + 9 + 3 \cdot 3 + 5 + 3 \cdot 7 + 3 + 3 \cdot 4 + 3 + 3 \cdot 1 + 0 + 3 \cdot 4 + x_{12} \equiv 0 \pmod{10}$
      2. $21 + 9 + 9 + 5 + 21 + 3 + 12 + 3 + 3 + 0 + 12 + x_{12} \equiv 0 \pmod{10}$

Checking digits: UPCs

1. A common method for detecting errors in strings of digits is to add an extra digit at the end, which is evaluated using a function.
2. If the final digit is not correct, then the string is assumed not to be correct.
3. Retail products are identified by their *Universal Product Codes* (*UPC*s). Usually these have 12 decimal digits, the last one being the check digit. The check digit x_{12} is determined by:
   $3x_1 + x_2 + 3x_3 + x_4 + 3x_5 + x_6 + 3x_7 + x_8 + 3x_9 + x_{10} + 3x_{11} + x_{12} \equiv 0 \mod 10$

   a. Suppose that the first 11 digits of the UPC are 79357343104. What is the check digit?
      1. $3 \cdot 7 + 9 + 3 \cdot 3 + 5 + 3 \cdot 7 + 3 + 3 \cdot 4 + 3 + 3 \cdot 1 + 0 + 3 \cdot 4 + x_{12} \equiv 0 \pmod{10}$
      2. $21 + 9 + 9 + 5 + 21 + 3 + 12 + 3 + 3 + 0 + 12 + x_{12} \equiv 0 \pmod{10}$
      3. $98 + x_{12} \equiv 0 \pmod{10}$

Checking digits: UPCs

1. A common method for detecting errors in strings of digits is to add an extra digit at the end, which is evaluated using a function.
2. If the final digit is not correct, then the string is assumed not to be correct.
3. Retail products are identified by their *Universal Product Codes* (*UPC*s). Usually these have 12 decimal digits, the last one being the check digit. The check digit x_{12} is determined by:
   $3x_1 + x_2 + 3x_3 + x_4 + 3x_5 + x_6 + 3x_7 + x_8 + 3x_9 + x_{10} + 3x_{11} + x_{12} \equiv 0 \mod 10$

   a. Suppose that the first 11 digits of the UPC are 79357343104. What is the check digit?
      1. $3 \cdot 7 + 9 + 3 \cdot 3 + 5 + 3 \cdot 7 + 3 + 3 \cdot 4 + 3 + 3 \cdot 1 + 0 + 3 \cdot 4 + x_{12} \equiv 0 \pmod{10}$
      2. $21 + 9 + 9 + 5 + 21 + 3 + 12 + 3 + 3 + 0 + 12 + x_{12} \equiv 0 \pmod{10}$
      3. $98 + x_{12} \equiv 0 \pmod{10}$
      4. So, the check digit is 2.

Checking digits: UPCs

1. A common method for detecting errors in strings of digits is to add an extra digit at the end, which is evaluated using a function.
2. If the final digit is not correct, then the string is assumed not to be correct.
3. Retail products are identified by their *Universal Product Codes* (*UPC*s). Usually these have 12 decimal digits, the last one being the check digit. The check digit x_{12} is determined by:
   $3x_1 + x_2 + 3x_3 + x_4 + 3x_5 + x_6 + 3x_7 + x_8 + 3x_9 + x_{10} + 3x_{11} + x_{12} \equiv 0 \mod 10$

   a. Suppose that the first 11 digits of the UPC are 79357343104. What is the check digit?
      1. $3 \cdot 7 + 9 + 3 \cdot 3 + 5 + 3 \cdot 7 + 3 + 3 \cdot 4 + 3 + 3 \cdot 1 + 0 + 3 \cdot 4 + x_{12} \equiv 0 \pmod{10}$
      2. $21 + 9 + 9 + 5 + 21 + 3 + 12 + 3 + 3 + 0 + 12 + x_{12} \equiv 0 \pmod{10}$
      3. $98 + x_{12} \equiv 0 \pmod{10}$
      4. So, the check digit is 2.

   b. Is 041331021641 a valid UPC?

Checking digits: UPCs

1. A common method for detecting errors in strings of digits is to add an extra digit at the end, which is evaluated using a function.
2. If the final digit is not correct, then the string is assumed not to be correct.
3. Retail products are identified by their *Universal Product Codes* (*UPC*s). Usually these have 12 decimal digits, the last one being the check digit. The check digit x_{12} is determined by:
   $3x_1 + x_2 + 3x_3 + x_4 + 3x_5 + x_6 + 3x_7 + x_8 + 3x_9 + x_{10} + 3x_{11} + x_{12} \equiv 0 \mod 10$

   a. Suppose that the first 11 digits of the UPC are 79357343104. What is the check digit?
      1. $3 \cdot 7 + 9 + 3 \cdot 3 + 5 + 3 \cdot 7 + 3 + 3 \cdot 4 + 3 + 3 \cdot 1 + 0 + 3 \cdot 4 + x_{12} \equiv 0 \pmod{10}$
      2. $21 + 9 + 9 + 5 + 21 + 3 + 12 + 3 + 3 + 0 + 12 + x_{12} \equiv 0 \pmod{10}$
      3. $98 + x_{12} \equiv 0 \pmod{10}$
      4. So, the check digit is 2.

   b. Is 041331021641 a valid UPC?
      1. $0 + 4 + 3 + 3 + 9 + 1 + 0 + 2 + 3 + 6 + 12 + 1 = 44 \not\equiv 0 \pmod{10}$

Checking digits: UPCs

1. A common method for detecting errors in strings of digits is to add an extra digit at the end, which is evaluated using a function.
2. If the final digit is not correct, then the string is assumed not to be correct.
3. Retail products are identified by their *Universal Product Codes* (*UPC*s). Usually these have 12 decimal digits, the last one being the check digit. The check digit x_{12} is determined by: $3x_1 + x_2 + 3x_3 + x_4 + 3x_5 + x_6 + 3x_7 + x_8 + 3x_9 + x_{10} + 3x_{11} + x_{12} \equiv 0 \mod 10$

   a. Suppose that the first 11 digits of the UPC are 79357343104. What is the check digit?
      1. $3\cdot7+9+3\cdot3+5+3\cdot7+3+3\cdot4+3+3\cdot1+0+3\cdot4+x_{12} \equiv 0 \pmod{10}$
      2. $21+9+9+5+21+3+12+3+3+0+12+x_{12} \equiv 0 \pmod{10}$
      3. $98+x_{12} \equiv 0 \pmod{10}$
      4. So, the check digit is 2.

   b. Is 041331021641 a valid UPC?
      1. $0+4+3+3+9+1+0+2+3+6+12+1 = 44 \not\equiv 0 \pmod{10}$
      2. Hence, 041331021641 is not a valid UPC.

Checking digits: ISBNs

1. **B**ooks are identified by an *International Standard Book Number* (ISBN-10), a 10 digit code

$$x_1, x_2, x_3, \ldots, x_9, x_{10}$$

Checking digits: ISBNs

1. **B**ooks are identified by an *International Standard Book Number* (ISBN-10), a 10 digit code

$$x_1, x_2, x_3, \ldots, x_9, x_{10}$$

2. The first 9 digits identify the language, the publisher, and the book. The tenth digit is a check digit, which is determined by the following congruence

$$x_{10} \equiv \sum_{i=1}^{9} i x_i \pmod{11}$$

Checking digits: ISBNs

1. **B**ooks are identified by an *International Standard Book Number* (ISBN-10), a 10 digit code

$$x_1, x_2, x_3, \ldots, x_9, x_{10}$$

2. The first 9 digits identify the language, the publisher, and the book. The tenth digit is a check digit, which is determined by the following congruence

$$x_{10} \equiv \sum_{i=1}^{9} i x_i \pmod{11}$$

3. Since $11x_{10} \equiv 0 \pmod{11}$ and $x_{10} + 10x_{10} \equiv \sum_{i=1}^{10} i x_i \pmod{11}$ it is easy to show that the validity of an ISBN-10 number can be equivalently evaluated by checking

$$\sum_{i=1}^{10} i x_i \equiv 0 \pmod{11}$$

Checking digits: ISBNs

$$x_{10} \equiv \sum_{i=1}^{9} i x_i \pmod{11} \Leftrightarrow \sum_{i=1}^{10} i x_i \equiv 0 \pmod{11}$$

1. Suppose that the first 9 digits of the ISBN-10 are 007288008. What is the check digit?

Checking digits: ISBNs

$$x_{10} \equiv \sum_{i=1}^{9} i x_i \pmod{11} \Leftrightarrow \sum_{i=1}^{10} i x_i \equiv 0 \pmod{11}$$

1. Suppose that the first 9 digits of the ISBN-10 are 007288008. What is the check digit?
   Solution:
   a. $x_{10} \equiv 1 \cdot 0 + 2 \cdot 0 + 3 \cdot 7 + 4 \cdot 2 + 5 \cdot 8 + 6 \cdot 8 + 7 \cdot 0 + 8 \cdot 0 + 9 \cdot 8$ mod 11.

Checking digits: ISBNs

$$x_{10} \equiv \sum_{i=1}^{9} ix_i \pmod{11} \Leftrightarrow \sum_{i=1}^{10} ix_i \equiv 0 \pmod{11}$$

1. Suppose that the first 9 digits of the ISBN-10 are 007288008. What is the check digit?
   Solution:
   a. $x_{10} \equiv 1 \cdot 0 + 2 \cdot 0 + 3 \cdot 7 + 4 \cdot 2 + 5 \cdot 8 + 6 \cdot 8 + 7 \cdot 0 + 8 \cdot 0 + 9 \cdot 8 \mod 11$.
   b. $x_{10} \equiv 0 + 0 + 21 + 8 + 40 + 48 + 0 + 0 + 72 \mod 11$.

Checking digits: ISBNs

$$x_{10} \equiv \sum_{i=1}^{9} ix_i \pmod{11} \Leftrightarrow \sum_{i=1}^{10} ix_i \equiv 0 \pmod{11}$$

1. Suppose that the first 9 digits of the ISBN-10 are 007288008. What is the check digit?
   Solution:
   a. $x_{10} \equiv 1 \cdot 0 + 2 \cdot 0 + 3 \cdot 7 + 4 \cdot 2 + 5 \cdot 8 + 6 \cdot 8 + 7 \cdot 0 + 8 \cdot 0 + 9 \cdot 8 \mod 11$.
   b. $x_{10} \equiv 0 + 0 + 21 + 8 + 40 + 48 + 0 + 0 + 72 \mod 11$.
   c. $x_{10} \equiv 189 \equiv 2 \mod 11$. Hence, $x_{10} = 2$.

Checking digits: ISBNs

$$x_{10} \equiv \sum_{i=1}^{9} ix_i \pmod{11} \Leftrightarrow \sum_{i=1}^{10} ix_i \equiv 0 \pmod{11}$$

1. Suppose that the first 9 digits of the ISBN-10 are 007288008. What is the check digit?
   Solution:
   a. $x_{10} \equiv 1 \cdot 0 + 2 \cdot 0 + 3 \cdot 7 + 4 \cdot 2 + 5 \cdot 8 + 6 \cdot 8 + 7 \cdot 0 + 8 \cdot 0 + 9 \cdot 8 \mod 11$.
   b. $x_{10} \equiv 0 + 0 + 21 + 8 + 40 + 48 + 0 + 0 + 72 \mod 11$.
   c. $x_{10} \equiv 189 \equiv 2 \mod 11$. Hence, $x_{10} = 2$.
2. Is 084930149X a valid ISBN10? (X is used as the digit 10.)

Checking digits: ISBNs

$$x_{10} \equiv \sum_{i=1}^{9} ix_i \pmod{11} \Leftrightarrow \sum_{i=1}^{10} ix_i \equiv 0 \pmod{11}$$

1. Suppose that the first 9 digits of the ISBN-10 are 007288008. What is the check digit?
   Solution:
   a. $x_{10} \equiv 1 \cdot 0 + 2 \cdot 0 + 3 \cdot 7 + 4 \cdot 2 + 5 \cdot 8 + 6 \cdot 8 + 7 \cdot 0 + 8 \cdot 0 + 9 \cdot 8 \mod 11$.
   b. $x_{10} \equiv 0 + 0 + 21 + 8 + 40 + 48 + 0 + 0 + 72 \mod 11$.
   c. $x_{10} \equiv 189 \equiv 2 \mod 11$. Hence, $x_{10} = 2$.
2. Is 084930149X a valid ISBN10? (X is used as the digit 10.)
   Solution:
   a. $1 \cdot 0 + 2 \cdot 8 + 3 \cdot 4 + 4 \cdot 9 + 5 \cdot 3 + 6 \cdot 0 + 7 \cdot 1 + 8 \cdot 4 + 9 \cdot 9 + 10 \cdot 10$

Checking digits: ISBNs

$$x_{10} \equiv \sum_{i=1}^{9} i x_i \pmod{11} \Leftrightarrow \sum_{i=1}^{10} i x_i \equiv 0 \pmod{11}$$

1. Suppose that the first 9 digits of the ISBN-10 are 007288008. What is the check digit?
   Solution:
   a. $x_{10} \equiv 1 \cdot 0 + 2 \cdot 0 + 3 \cdot 7 + 4 \cdot 2 + 5 \cdot 8 + 6 \cdot 8 + 7 \cdot 0 + 8 \cdot 0 + 9 \cdot 8$ mod 11.
   b. $x_{10} \equiv 0 + 0 + 21 + 8 + 40 + 48 + 0 + 0 + 72$ mod 11.
   c. $x_{10} \equiv 189 \equiv 2$ mod 11. Hence, $x_{10} = 2$.
2. Is 084930149X a valid ISBN10? (X is used as the digit 10.)
   Solution:
   a. $1 \cdot 0 + 2 \cdot 8 + 3 \cdot 4 + 4 \cdot 9 + 5 \cdot 3 + 6 \cdot 0 + 7 \cdot 1 + 8 \cdot 4 + 9 \cdot 9 + 10 \cdot 10$
   b. $= 0 + 16 + 12 + 36 + 15 + 0 + 7 + 32 + 81 + 100 = 299 \equiv 2 \not\equiv 0$ mod 11

Checking digits: ISBNs

$$x_{10} \equiv \sum_{i=1}^{9} ix_i \pmod{11} \Leftrightarrow \sum_{i=1}^{10} ix_i \equiv 0 \pmod{11}$$

1. Suppose that the first 9 digits of the ISBN-10 are 007288008. What is the check digit?
   Solution:
   a. $x_{10} \equiv 1 \cdot 0 + 2 \cdot 0 + 3 \cdot 7 + 4 \cdot 2 + 5 \cdot 8 + 6 \cdot 8 + 7 \cdot 0 + 8 \cdot 0 + 9 \cdot 8$ mod 11.
   b. $x_{10} \equiv 0 + 0 + 21 + 8 + 40 + 48 + 0 + 0 + 72$ mod 11.
   c. $x_{10} \equiv 189 \equiv 2$ mod 11. Hence, $x_{10} = 2$.
2. Is 084930149X a valid ISBN10? (X is used as the digit 10.)
   Solution:
   a. $1 \cdot 0 + 2 \cdot 8 + 3 \cdot 4 + 4 \cdot 9 + 5 \cdot 3 + 6 \cdot 0 + 7 \cdot 1 + 8 \cdot 4 + 9 \cdot 9 + 10 \cdot 10$
   b. $= 0 + 16 + 12 + 36 + 15 + 0 + 7 + 32 + 81 + 100 = 299 \equiv 2 \not\equiv 0$ mod 11
   c. Hence, 084930149X is not a valid ISBN-10.

Checking digits: ISBNs

$$x_{10} \equiv \sum_{i=1}^{9} ix_i \pmod{11} \Leftrightarrow \sum_{i=1}^{10} ix_i \equiv 0 \pmod{11}$$

1. Suppose that the first 9 digits of the ISBN-10 are 007288008. What is the check digit?
   Solution:
   a. $x_{10} \equiv 1 \cdot 0 + 2 \cdot 0 + 3 \cdot 7 + 4 \cdot 2 + 5 \cdot 8 + 6 \cdot 8 + 7 \cdot 0 + 8 \cdot 0 + 9 \cdot 8 \mod 11$.
   b. $x_{10} \equiv 0 + 0 + 21 + 8 + 40 + 48 + 0 + 0 + 72 \mod 11$.
   c. $x_{10} \equiv 189 \equiv 2 \mod 11$. Hence, $x_{10} = 2$.
2. Is 084930149X a valid ISBN10? (X is used as the digit 10.)
   Solution:
   a. $1 \cdot 0 + 2 \cdot 8 + 3 \cdot 4 + 4 \cdot 9 + 5 \cdot 3 + 6 \cdot 0 + 7 \cdot 1 + 8 \cdot 4 + 9 \cdot 9 + 10 \cdot 10$
   b. $= 0 + 16 + 12 + 36 + 15 + 0 + 7 + 32 + 81 + 100 = 299 \equiv 2 \not\equiv 0 \mod 11$
   c. Hence, 084930149X is not a valid ISBN-10.

A *single error* is an error in one digit of an identification number and a *transposition error* is the accidental interchanging of two digits. Both of these kinds of errors can be detected by the check digit for ISBN-10.

Plan for Part II

1. Solving Congruences
1.1 Linear Congruences
1.2 Systems of Linear Congruences

2. Applications of Congruences
2.1 Hashing Functions
2.2 Pseudorandom Numbers
2.3 Checking Digits

3. Cryptography
3.1 Classical cryptography
3.2 Public Key Cryptography
3.3 The RSA Encryption

Plan for Part II

Caesar cipher

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Julius Caesar created secret messages by shifting each letter three letters forward in the alphabet (sending the last three letters to the first three letters.)

Caesar cipher

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Julius Caesar created secret messages by shifting each letter three letters forward in the alphabet (sending the last three letters to the first three letters.) For example, the letter B is replaced by E and the letter X is replaced by A.

Caesar cipher

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Julius Caesar created secret messages by shifting each letter three letters forward in the alphabet (sending the last three letters to the first three letters.) For example, the letter B is replaced by E and the letter X is replaced by A. This process of making a message secret is an example of *encryption*.

Caesar cipher

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Julius Caesar created secret messages by shifting each letter three letters forward in the alphabet (sending the last three letters to the first three letters.) For example, the letter B is replaced by E and the letter X is replaced by A. This process of making a message secret is an example of *encryption*. Here is how the encryption process works:

Caesar cipher

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Julius Caesar created secret messages by shifting each letter three letters forward in the alphabet (sending the last three letters to the first three letters.) For example, the letter B is replaced by E and the letter X is replaced by A. This process of making a message secret is an example of *encryption*. Here is how the encryption process works:

1. Replace each letter by an integer from $\mathbb{Z}_{26}$, that is an integer from 0 to 25 representing one less than its position in the alphabet.

Caesar cipher

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Julius Caesar created secret messages by shifting each letter three letters forward in the alphabet (sending the last three letters to the first three letters.) For example, the letter B is replaced by E and the letter X is replaced by A. This process of making a message secret is an example of *encryption*. Here is how the encryption process works:

1. Replace each letter by an integer from $\mathbb{Z}_{26}$, that is an integer from 0 to 25 representing one less than its position in the alphabet.
2. The encryption function is $f(p) = (p + 3) \pmod{26}$. It replaces each integer p in the set $\{0, 1, 2, \ldots, 25\}$ by $f(p)$ in the set $\{0, 1, 2, \ldots, 25\}$.

Caesar cipher

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Julius Caesar created secret messages by shifting each letter three letters forward in the alphabet (sending the last three letters to the first three letters.) For example, the letter B is replaced by E and the letter X is replaced by A. This process of making a message secret is an example of *encryption*. Here is how the encryption process works:

1. Replace each letter by an integer from $\mathbb{Z}_{26}$, that is an integer from 0 to 25 representing one less than its position in the alphabet.
2. The encryption function is $f(p) = (p+3) \pmod{26}$. It replaces each integer p in the set $\{0, 1, 2, \ldots, 25\}$ by $f(p)$ in the set $\{0, 1, 2, \ldots, 25\}$.
3. Replace each integer p by the letter with the position $p+1$ in the alphabet.

Caesar cipher

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Example

Encrypt the message “MEET YOU IN THE PARK” using the Caesar cipher.

Caesar cipher

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Example

Encrypt the message "MEET YOU IN THE PARK" using the Caesar cipher.

1. Write with numbers in $\mathbb{Z}_{26}$: 12 4 4 19 24 14 20 8 13 19 7 4 15 0 17 10.

Caesar cipher

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Example

Encrypt the message "MEET YOU IN THE PARK" using the Caesar cipher.

1. Write with numbers in $\mathbb{Z}_{26}$: 12 4 4 19 24 14 20 8 13 19 7 4 15 0 17 10.
2. Now replace each of these numbers p by $f(p) = (p+3) \pmod{26}$.

Caesar cipher

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Example

Encrypt the message “MEET YOU IN THE PARK” using the Caesar cipher.

1. Write with numbers in $\mathbb{Z}_{26}$: 12 4 4 19 24 14 20 8 13 19 7 4 15 0 17 10.
2. Now replace each of these numbers p by $f(p) = (p + 3) \pmod{26}$.
3. 15 7 7 22 1 17 23 11 16 22 10 7 18 3 20 13.

Caesar cipher

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Example

Encrypt the message "MEET YOU IN THE PARK" using the Caesar cipher.

1. Write with numbers in $\mathbb{Z}_{26}$: 12 4 4 19 24 14 20 8 13 19 7 4 15 0 17 10.
2. Now replace each of these numbers p by $f(p) = (p + 3) \pmod{26}$.
3. 15 7 7 22 1 17 23 11 16 22 10 7 18 3 20 13.
4. Translating the numbers back to letters produces the encrypted message "PHHW BRX LQ WKH SDUN."

Caesar cipher

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

1. To recover the original message, use $f^{-1}(p) = (p - 3) \pmod{26}$. So, each letter in the coded message is shifted back three letters in the alphabet, with the first three letters sent to the last three letters.

Caesar cipher

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

1. To recover the original message, use $f^{-1}(p) = (p - 3) \pmod{26}$. So, each letter in the coded message is shifted back three letters in the alphabet, with the first three letters sent to the last three letters.
2. This process of recovering the original message from the encrypted message is called *decryption* .

Caesar cipher

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

1. To recover the original message, use $f^{-1}(p) = (p-3) \pmod{26}$. So, each letter in the coded message is shifted back three letters in the alphabet, with the first three letters sent to the last three letters.
2. This process of recovering the original message from the encrypted message is called *decryption* .
3. The Caesar cipher is one of a family of ciphers called *shift ciphers.* Letters can be shifted by an integer k, with 3 being just one possibility. The encryption function is
   a. $f(p) = (p+k) \pmod{26}$

Caesar cipher

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

1. To recover the original message, use $f^{-1}(p) = (p - 3) \pmod{26}$. So, each letter in the coded message is shifted back three letters in the alphabet, with the first three letters sent to the last three letters.
2. This process of recovering the original message from the encrypted message is called *decryption* .
3. The Caesar cipher is one of a family of ciphers called *shift ciphers.* Letters can be shifted by an integer k, with 3 being just one possibility. The encryption function is
   a. $f(p) = (p + k) \pmod{26}$

   and the decryption function is

Caesar cipher

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

1. To recover the original message, use $f^{-1}(p) = (p - 3) \pmod{26}$. So, each letter in the coded message is shifted back three letters in the alphabet, with the first three letters sent to the last three letters.
2. This process of recovering the original message from the encrypted message is called *decryption* .
3. The Caesar cipher is one of a family of ciphers called *shift ciphers.* Letters can be shifted by an integer k, with 3 being just one possibility. The encryption function is
   a. $f(p) = (p + k) \pmod{26}$

   and the decryption function is
   b. $f^{-1}(p) = (p - k) \pmod{26}$

Caesar cipher

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

1. To recover the original message, use $f^{-1}(p) = (p - 3) \pmod{26}$. So, each letter in the coded message is shifted back three letters in the alphabet, with the first three letters sent to the last three letters.
2. This process of recovering the original message from the encrypted message is called *decryption* .
3. The Caesar cipher is one of a family of ciphers called *shift ciphers.* Letters can be shifted by an integer k, with 3 being just one possibility. The encryption function is
   a. $f(p) = (p + k) \pmod{26}$

   and the decryption function is

   b. $f^{-1}(p) = (p - k) \pmod{26}$
4. The integer k is called a *key*.

Shift cipher

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Example

Encrypt the message "STOP GLOBAL WARMING" using the shift cipher with $k = 11$.

Solution :

Shift cipher

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Example

Encrypt the message "STOP GLOBAL WARMING" using the shift cipher with $k = 11$.

Solution :

1. Replace each letter with the corresponding element of $\mathbb{Z}_{26}$.

   18 19 14 15 6 11 14 1 0 11 22 0 17 12 8 13 6.

Shift cipher

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Example

Encrypt the message "STOP GLOBAL WARMING" using the shift cipher with $k = 11$.

Solution :

1. Replace each letter with the corresponding element of $\mathbb{Z}_{26}$.
   18 19 14 15 6 11 14 1 0 11 22 0 17 12 8 13 6.
2. Apply the shift $f(p) = (p + 11) \pmod{26}$, yielding
   3 4 25 0 17 22 25 12 11 22 7 11 2 23 19 24 17.

Shift cipher

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Example

Encrypt the message "STOP GLOBAL WARMING" using the shift cipher with $k = 11$.

Solution :

1. Replace each letter with the corresponding element of $\mathbb{Z}_{26}$.

   18 19 14 15 6 11 14 1 0 11 22 0 17 12 8 13 6.

2. Apply the shift $f(p) = (p + 11) \pmod{26}$, yielding

   3 4 25 0 17 22 25 12 11 22 7 11 2 23 19 24 17.

3. Translating the numbers back to letters produces the ciphertext

   "DEZA RWZMLW HLCXTYR."

Shift cipher

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Example

Decrypt the message "LEWLYPLUJL PZ H NYLHA ALHJOLY" that was encrypted using the shift cipher with $k = 7$.

Solution:

Shift cipher

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Example

Decrypt the message "LEWLYPLUJL PZ H NYLHA ALHJOLY" that was encrypted using the shift cipher with $k = 7$.

Solution:

1. Replace each letter with the corresponding element of $\mathbb{Z}_{26}$.

   11 4 22 11 24 15 11 20 9 11 15 25 7 13 24 11 7 0 0 11 7 9 14 11 24.

Shift cipher

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Example

Decrypt the message "LEWLYPLUJL PZ H NYLHA ALHJOLY" that was encrypted using the shift cipher with $k = 7$.

Solution:

1. Replace each letter with the corresponding element of $\mathbb{Z}_{26}$.

   11 4 22 11 24 15 11 20 9 11 15 25 7 13 24 11 7 0 0 11 7 9 14 11 24.

2. Shift each of the numbers by $-k = -7$ modulo 26, yielding

   4 23 15 4 17 8 4 13 2 4 8 18 0 6 17 4 0 19 19 4 0 2 7 4 17.

Shift cipher

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Example

Decrypt the message "LEWLYPLUJL PZ H NYLHA ALHJOLY" that was encrypted using the shift cipher with $k = 7$.

Solution:

1. Replace each letter with the corresponding element of $\mathbb{Z}_{26}$.

   11 4 22 11 24 15 11 20 9 11 15 25 7 13 24 11 7 0 0 11 7 9 14 11 24.

2. Shift each of the numbers by $-k = -7$ modulo 26, yielding

   4 23 15 4 17 8 4 13 2 4 8 18 0 6 17 4 0 19 19 4 0 2 7 4 17.

3. Translating the numbers back to letters produces the decrypted message

   "EXPERIENCE IS A GREAT TEACHER."

Affine ciphers

Shift ciphers are a special case of *affine ciphers* which use functions of the form

$$f(p) = (ap + b) \pmod{26}$$

where a and b are integers, chosen so that f is a bijection.

Affine ciphers

Shift ciphers are a special case of *affine ciphers* which use functions of the form

$$f(p) = (ap + b) \pmod{26}$$

where a and b are integers, chosen so that f is a bijection.

Note: this function is a bijection if and only if $\gcd(a, 26) = 1$*. See Tutorial 7.*

Affine ciphers

Shift ciphers are a special case of *affine ciphers* which use functions of the form

$$f(p) = (ap + b) \pmod{26}$$

where a and b are integers, chosen so that f is a bijection.

Note: this function is a bijection if and only if $\gcd(a, 26) = 1$. *See Tutorial 7.*

Example

What letter replaces the letter K when the function $f(p) = (7p + 3) \pmod{26}$ is used for encryption.

Affine ciphers

Shift ciphers are a special case of *affine ciphers* which use functions of the form

$$f(p) = (ap + b) \pmod{26}$$

where a and b are integers, chosen so that f is a bijection.

Note: this function is a bijection if and only if $\gcd(a, 26) = 1$. *See Tutorial 7.*

Example

What letter replaces the letter K when the function $f(p) = (7p + 3) \pmod{26}$ is used for encryption.

Solution : Since 10 represents K, $f(10) = (7 \cdot 10 + 3) \pmod{26} = 21$, which corresponds to letter V.

Affine ciphers

To decrypt a message encrypted by a shift cipher, the congruence $c \equiv ap + b \pmod{26}$ needs to be solved for p.

Affine ciphers

To decrypt a message encrypted by a shift cipher, the congruence $c \equiv ap + b \pmod{26}$ needs to be solved for p.

1. Subtract b from both sides to obtain
$$ap \equiv c - b \pmod{26}$$

Affine ciphers

To decrypt a message encrypted by a shift cipher, the congruence $c \equiv ap + b \pmod{26}$ needs to be solved for p.

1. Subtract b from both sides to obtain
$$ap \equiv c - b \pmod{26}$$
2. Multiply both sides by the inverse $\overline{a}$ of a modulo 26, which exists since $\gcd(a, 26) = 1$
$$\overline{a}ap \equiv \overline{a}(c - b) \pmod{26}$$

Affine ciphers

To decrypt a message encrypted by a shift cipher, the congruence $c \equiv ap + b \pmod{26}$ needs to be solved for p.

1. Subtract b from both sides to obtain
$$ap \equiv c - b \pmod{26}$$
2. Multiply both sides by the inverse $\overline{a}$ of a modulo 26, which exists since $\gcd(a, 26) = 1$
$$\overline{a}ap \equiv \overline{a}(c - b) \pmod{26}$$
which simplifies to
$$p \equiv \overline{a}(c - b) \pmod{26}$$
determining p in $\mathbb{Z}_{26}$ given a, b and cryptotext c.

Example

Example

1. What is the decryption function for an affine cipher $f(x) \equiv 3x + 7 \pmod{26}$?

Example

Example

1. What is the decryption function for an affine cipher $f(x) \equiv 3x + 7 \pmod{26}$?

   Solution: $f^{-1}(x) \equiv 9x + 15 \pmod{26}$

   Note: 9 is inverse of 3 modulo 26 and
   $-9 \cdot 7 = -63 \equiv 15 \pmod{26}$

Example

Example

1. What is the decryption function for an affine cipher $f(x) \equiv 3x + 7 \pmod{26}$?

   Solution: $f^{-1}(x) \equiv 9x + 15 \pmod{26}$

   Note: 9 is inverse of 3 modulo 26 and $-9 \cdot 7 = -63 \equiv 15 \pmod{26}$

2. Decrypt the following message encrypted by the above "UTTQ CTOA"

Example

Example

1. What is the decryption function for an affine cipher $f(x) \equiv 3x + 7 \pmod{26}$?

   Solution: $f^{-1}(x) \equiv 9x + 15 \pmod{26}$

   Note: 9 is inverse of 3 modulo 26 and $-9 \cdot 7 = -63 \equiv 15 \pmod{26}$

2. Decrypt the following message encrypted by the above "UTTQ CTOA"

   Solution: "NEED HELP"

Plan for Part II

Public key cryptography

1. All classical ciphers, including shift and affine ciphers, are *private key cryptosystems*. Knowing the encryption key allows one to quickly determine the decryption key.

Public key cryptography

1. All classical ciphers, including shift and affine ciphers, are *private key cryptosystems*. Knowing the encryption key allows one to quickly determine the decryption key.
2. All parties who wish to communicate using a private key cryptosystem must share the key and keep it a secret.

Public key cryptography

1. All classical ciphers, including shift and affine ciphers, are *private key cryptosystems*. Knowing the encryption key allows one to quickly determine the decryption key.
2. All parties who wish to communicate using a private key cryptosystem must share the key and keep it a secret.
3. In public key cryptosystems, first invented in the 1970s, knowing how to encrypt a message does not help one to decrypt the message.

Public key cryptography

1. All classical ciphers, including shift and affine ciphers, are *private key cryptosystems*. Knowing the encryption key allows one to quickly determine the decryption key.
2. All parties who wish to communicate using a private key cryptosystem must share the key and keep it a secret.
3. In public key cryptosystems, first invented in the 1970s, knowing how to encrypt a message does not help one to decrypt the message.
4. Therefore, everyone can have a publicly known encryption key. The only key that needs to be kept secret is the decryption key.

Plan for Part II

The RSA Cryptosystem

Clifford Cocks (Born 1950)

1. A public key cryptosystem, now known as the RSA system was introduced in 1976 by three researchers at MIT.

Ronald Rivest (Born 1948)

Adi Shamir (Born 1952)

Leonard Adelman (Born 1945)

It is now known that the method was discovered earlier by Clifford Cocks, working secretly for the UK government.

The RSA Cryptosystem

Clifford Cocks (Born 1950)

1. A public key cryptosystem, now known as the RSA system was introduced in 1976 by three researchers at MIT.

Ronald Rivest (Born 1948)

Adi Shamir (Born 1952)

Leonard Adelman (Born 1945)

It is now known that the method was discovered earlier by Clifford Cocks, working secretly for the UK government.

2. The public encryption key is a pair (n, e) where the modulus n is the product of two large (200 digits) primes p and q and exponent e is relatively prime to $(p - 1)(q - 1)$.

The RSA Cryptosystem

Clifford Cocks (Born 1950)

1. A public key cryptosystem, now known as the RSA system was introduced in 1976 by three researchers at MIT.

Ronald Rivest (Born 1948)

Adi Shamir (Born 1952)

Leonard Adelman (Born 1945)

It is now known that the method was discovered earlier by Clifford Cocks, working secretly for the UK government.

2. The public encryption key is a pair (n, e) where the modulus n is the product of two large (200 digits) primes p and q and exponent e is relatively prime to $(p-1)(q-1)$.

3. Factorization $n = p \cdot q$ is kept private! With approximately 400 digits, n cannot be factored in a reasonable length of time.

The RSA encryption (overview)

To encrypt a message using RSA using a public key (n, e):

The RSA encryption (overview)

To encrypt a message using RSA using a public key (n, e):

1. Translate the *plain text message M* into sequences of two digit integers representing the letters. Use 00 for A, 01 for B, etc.

The RSA encryption (overview)

To encrypt a message using RSA using a public key (n, e):

1. Translate the *plain text message M* into sequences of two digit integers representing the letters. Use 00 for A, 01 for B, etc.
2. Concatenate the two digit integers into strings of digits.

The RSA encryption (overview)

To encrypt a message using RSA using a public key (n, e):

1. Translate the *plain text message M* into sequences of two digit integers representing the letters. Use 00 for A, 01 for B, etc.
2. Concatenate the two digit integers into strings of digits.
3. Divide this string into equally sized blocks of $2N$ digits where $2N$ is the largest even number with $2N$ digits that does not exceed n.

The RSA encryption (overview)

To encrypt a message using RSA using a public key (n, e):

1. Translate the *plain text message M* into sequences of two digit integers representing the letters. Use 00 for A, 01 for B, etc.
2. Concatenate the two digit integers into strings of digits.
3. Divide this string into equally sized blocks of $2N$ digits where $2N$ is the largest even number with $2N$ digits that does not exceed n.
4. The plain text message M is now a sequence of integers $m_1, m_2, \ldots, m_k$.

The RSA encryption (overview)

To encrypt a message using RSA using a public key (n, e):

1. Translate the *plain text message M* into sequences of two digit integers representing the letters. Use 00 for A, 01 for B, etc.
2. Concatenate the two digit integers into strings of digits.
3. Divide this string into equally sized blocks of $2N$ digits where $2N$ is the largest even number with $2N$ digits that does not exceed n.
4. The plain text message M is now a sequence of integers $m_1, m_2, \ldots, m_k$.
5. Each block (an integer) is encrypted using modular exponentiation function (efficiently computable, see Tutorial 7) that gives *ciphertext message C*:

$$C = M^e \pmod{n}$$

The RSA decryption (overview)

1. Decryption $C \to M$ requires known *exponentiation inverse* d of e modulo n:
$$C^d = (M^e)^d \equiv M \pmod{n}$$
Modular exponentiation is a *one-way function* : it is easy to compute , but hard to invert. In general, finding modular exponential inverse d is believed to be very difficult (as difficult as finding prime factorization of modulus n).

2. RSA assumes "privately" known factorization $n = p \cdot q$ where p and q are prime. **In this case**, the decryption key d can be obtained as a multiplicative inverse of e modulo $(p-1)(q-1)$, which is easy to compute (via Euclidean algorithm for Bézout coefficients) assuming relative primality $\gcd(e,(p-1)(q-1)) = 1$. It can be shown that such (privately known) key d allows to decrypt ciphertext message C with the simple computation:
$$M = C^d \mod (p \cdot q)$$

3. RSA works as a public key system since the only known method of finding d is based on a factorization of n into primes. There is currently no known feasible method for factoring large numbers into primes.