

# Discrete Mathematics for Computer Science

## Lecture 22: Review

Dr. Ming Tang

Department of Computer Science and Engineering  
Southern University of Science and Technology (SUSTech)  
Email: tangm3@sustech.edu.cn

# Lecture Schedule

- |                                  |             |
|----------------------------------|-------------|
| 1 Logic and Mathematical Proofs  | 6 Recursion |
| 2 Sets and Functions             | 7 Counting  |
| 3 Complexity of Algorithms       | 8 Relations |
| 4 Number Theory and Cryptography | 9 Graph     |
| 5 Mathematical Induction         | 10 Trees    |

# Division

**Divisibility:** We say that  $a$  divides  $b$  if there is an integer  $c$  such that  $b = ac$ , or equivalently  $b/a$  is an integer.

- If  $a, b, c$  are integers, where  $a \neq 0$ , such that  $a|b$  and  $a|c$ , then  $a|(mb + nc)$  whenever  $m$  and  $n$  are integers.

**Congruence Relation:** If  $a$  and  $b$  are integers and  $m$  is a positive integer, then  $a$  is congruent to  $b$  modulo  $m$  if  $m$  divides  $a - b$ , denoted by  $a \equiv b \pmod{m}$ . Equivalently,

$$a \bmod m = b \bmod m$$

The integers  $a$  and  $b$  are congruent modulo  $m$  if and only if there is an integer  $k$  such that

$$a = b + km$$

# Congruence: Properties

**Theorem:** Let  $m$  be a positive integer. If  $a \equiv b \pmod{m}$  and  $c \equiv d \pmod{m}$ , then

$$a + c \equiv b + d \pmod{m}$$

$$ac \equiv bd \pmod{m}$$

**Corollary:** Let  $m$  be a positive integer and let  $a$  and  $b$  be integers. Then,

$$(a + b) \pmod{m} = ((a \pmod{m}) + (b \pmod{m})) \pmod{m}$$

$$ab \pmod{m} = ((a \pmod{m})(b \pmod{m})) \pmod{m}$$

How to compute  $b^n \pmod{m}$ ?

If  $ca \equiv cb \pmod{m}$ , then  $a \equiv b \pmod{m}$ ? Any condition?



**SUSTech**

Southern University  
of Science and  
Technology

# Primes

An integer  $p$  that is greater than 1 is called a **prime** if the **only** positive factors of  $p$  are 1 and  $p$ .

- If  $n$  is composite, then  $n$  has a prime divisor less than or equal to  $\sqrt{n}$ .

Let  $a$  and  $b$  be integers, not both 0. The **largest** integer  $d$  such that  $d|a$  and  $d|b$  is called the **greatest common divisor** of  $a$  and  $b$ , denoted by **gcd**( $a, b$ ). Let  $a = p_1^{a_1} p_2^{a_2} \dots p_n^{a_n}$  and  $b = p_1^{b_1} p_2^{b_2} \dots p_n^{b_n}$ . Then,

$$\text{gcd}(a, b) = p^{\min(a_1, b_1)} p^{\min(a_2, b_2)} \dots p^{\min(a_n, b_n)}$$

The **least common multiple** of  $a$  and  $b$  is the **smallest positive integer** that is divisible by both  $a$  and  $b$ , denoted by  $\text{lcm}(a, b)$ . Let  $a = p_1^{a_1} p_2^{a_2} \dots p_n^{a_n}$  and  $b = p_1^{b_1} p_2^{b_2} \dots p_n^{b_n}$ . Then,

$$\text{lcm}(a, b) = p^{\max(a_1, b_1)} p^{\max(a_2, b_2)} \dots p^{\max(a_n, b_n)}.$$

# Euclidean Algorithm

Computing the **greatest common divisor** of two integers directly from the prime factorizations can be **time consuming** since we need to find all factors of the two integers.

For two integers 287 and 91, we want to find  $\gcd(287, 91)$ .

$$\text{Step 1: } 287 = 91 \cdot 3 + 14$$

$$\text{Step 2: } 91 = 14 \cdot 6 + 7$$

$$\text{Step 3: } 14 = 7 \cdot 2 + 0$$

$$\gcd(287, 91) = \gcd(91, 14) = \gcd(14, 7) = 7$$

How to prove?

# GCD as Linear Combinations

**Bezout's Theorem:** If  $a$  and  $b$  are positive integers, then there exist integers  $s$  and  $t$  such that

$$\gcd(a, b) = sa + tb.$$

This equation is called Bezout's identity.

We can use **extended Euclidean algorithm** to find Bezout's identity.

**Lemma:** If  $a$ ,  $b$ ,  $c$  are positive integers such that  $\gcd(a, b) = 1$  and  $a|bc$ , then  $a|c$ .

**Lemma:** If  $p$  is prime and  $p|a_1a_2\dots a_n$ , then  $p|a_i$  for some  $i$ .

# Linear Congruences

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.

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

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

$$x \equiv \bar{a}b \pmod{m}.$$



# Modular Inverse

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

When does inverse exist?

**Theorem:** If  $a$  and  $m$  are **relatively prime integers** and  $m > 1$ , then an inverse of  $a$  modulo  $m$  **exists**. The inverse is **unique** modulo  $m$ . That is,

- there is a unique positive integer  $\bar{a}$  less than  $m$  that is an inverse of  $a$  modulo  $m$  and
- every other inverse of  $a$  modulo  $m$  is congruent to  $\bar{a}$  modulo  $m$ .

If we obtain an arbitrary inverse of  $a$  modulo  $m$ , how to obtain the inverse that is less than  $m$ ?

# Modular Inverse

## How to find inverses?

Using **extended Euclidean algorithm**:

**Example:** Find an inverse of 101 modulo 4620. That is, find  $\bar{a}$  such that  $\bar{a} \cdot 101 \equiv 1 \pmod{4620}$ .

With extended Euclidean algorithm, we obtain  $\gcd(a, b) = sa + tb$ , i.e.,  $1 = -35 \cdot 4620 + 1601 \cdot 101$ . It tells us that  $-35$  and  $1601$  are Bezout coefficients of 4620 and 101. We have

$$1 \pmod{4620} = 1601 \cdot 101 \pmod{4620}.$$

Thus, 1601 is an inverse of 101 modulo 4620.

# The Chinese Remainder Theorem

**Theorem** (The Chinese Remainder Theorem): Let  $m_1, m_2, \dots, m_n$  be **pairwise relative prime** positive integers greater than 1 and  $a_1, a_2, \dots, a_n$  arbitrary integers. Then, the system

$$x \equiv a_1 \pmod{m_1}$$

$$x \equiv a_2 \pmod{m_2}$$

...

$$x \equiv a_n \pmod{m_n}$$

has a **unique solution** modulo  $m = m_1 m_2 \dots m_n$ .

(That is, there is a solution  $x$  with  $0 \leq x < m$ , and all other solutions are congruent modulo  $m$  to this solution.)

# The Chinese Remainder Theorem: Example

$$x \equiv 2 \pmod{3}$$

$$x \equiv 3 \pmod{5}$$

$$x \equiv 2 \pmod{7}$$

- 1 Let  $m = 3 \cdot 5 \cdot 7 = 105$ ,  $M_1 = m/3 = 35$ ,  $M_2 = m/5 = 21$ , and  $M_3 = m/7 = 15$ .
- 2 Compute  $y_k$ , i.e., the inverse of  $M_k$  modulo  $m_k$ :
  - ▶  $35 \cdot 2 \equiv 1 \pmod{3}$   $y_1 = 2$
  - ▶  $21 \equiv 1 \pmod{5}$   $y_2 = 1$
  - ▶  $15 \equiv 1 \pmod{7}$   $y_3 = 1$
- 3 Compute a solution  $x = a_1 M_1 y_1 + \dots + a_n M_n y_n$ :  
 $x = 2 \cdot 35 \cdot 2 + 3 \cdot 21 \cdot 1 + 2 \cdot 15 \cdot 1 \equiv 233 \equiv 23 \pmod{105}$
- 4 The solutions are all integers  $x$  that satisfy  $x \equiv 23 \pmod{105}$ .

How to prove  $x \equiv 23 \pmod{105}$  is the solution?

## Back Substitution

We may also solve systems of linear congruences with pairwise relatively prime moduli  $m_1, m_2, \dots, m_n$  by back substitution.

**Example:**

$$(1) \ x \equiv 1 \pmod{5}$$

$$(2) \ x \equiv 2 \pmod{6}$$

$$(3) \ x \equiv 3 \pmod{7}$$

According to (1),  $x = 5t + 1$ , where  $t$  is an integer.

Substituting this expression into (2), we have  $5t + 1 \equiv 2 \pmod{6}$ , which means that  $t \equiv 5 \pmod{6}$ . Thus,  $t = 6u + 5$ , where  $u$  is an integer.

Substituting  $x = 5t + 1$  and  $t = 6u + 5$  into (3), we have  $30u + 26 \equiv 3 \pmod{7}$ , which implies that  $u \equiv 6 \pmod{7}$ . Thus,  $u = 7v + 6$ , where  $v$  is an integer.

Thus, we must have  $x = 210v + 206$ . Translating this back into a congruence,

$$x \equiv 206 \pmod{210}.$$

# The Chinese Remainder Theorem

What if  $m_1, m_2, \dots, m_n$  are positive integers greater than 1, but they are **not** pairwise relatively prime?

$$x \equiv a_1 \pmod{m_1}$$

$$x \equiv a_2 \pmod{m_2}$$

...

$$x \equiv a_n \pmod{m_n}$$

**Translate** these congruence into a set of congruence that together are equivalent to the given congruence:

For  $x \equiv a_k \pmod{m_k}$ , suppose  $m_k$  can be written as  $m_k = b_k^1 b_k^2 \cdots b_k^r$ , where  $b_k^1, b_k^2, \dots, b_k^r$  are all primes. Then,  $x \equiv a_k \pmod{m_k}$  is equivalent to the following set of congruence:

$$x \equiv a_k \pmod{b_k^1}$$

$$x \equiv a_k \pmod{b_k^2}$$

...

$$x \equiv a_k \pmod{b_k^r}$$

# Fermat's Little Theorem

**FERMAT'S LITTLE THEOREM** If  $p$  is prime and  $a$  is an integer not divisible by  $p$ , then

$$a^{p-1} \equiv 1 \pmod{p}.$$

Furthermore, for every integer  $a$  we have

$$a^p \equiv a \pmod{p}.$$

# RSA Cryptosystem

Pick two large primes  $p$  and  $q$ . Let  $n = pq$ . **Encryption key**  $(n, e)$  and **decryption key**  $(n, d)$  are selected such that

$$(1) \gcd(e, (p-1)(q-1)) = 1$$

$$(2) ed \equiv 1 \pmod{(p-1)(q-1)}$$

**RSA encryption:**  $C = M^e \bmod n$ ;

**RSA decryption:**  $M = C^d \bmod n$ .

How to prove RSA decryption can recover the message?

Digital Signature?



# Lecture Schedule

- |                                  |             |
|----------------------------------|-------------|
| 1 Logic and Mathematical Proofs  | 6 Recursion |
| 2 Sets and Functions             | 7 Counting  |
| 3 Complexity of Algorithms       | 8 Relations |
| 4 Number Theory and Cryptography | 9 Graph     |
| 5 <b>Mathematical Induction</b>  | 10 Trees    |

# The Principle of Mathematical Induction

**Well-Ordering Property:** Every nonempty set of nonnegative integers has a least element.

## Principle. (Weak Principle of Mathematical Induction)

(a) **Basic Step:** the statement  $P(b)$  is true

(b) **Inductive Step:** the statement  $P(n-1) \rightarrow P(n)$  is true for all  $n > b$

Thus,  $P(n)$  is true for all integers  $n \geq b$ .

## Principle (Strong Principle of Mathematical Induction):

(a) **Basic Step:** the statement  $P(b)$  is true

(b) **Inductive Step:** for all  $n > b$ , the statement

$$P(b) \wedge P(b+1) \wedge \dots \wedge P(n-1) \rightarrow P(n) \text{ is true.}$$

Then,  $P(n)$  is true for all integers  $n \geq b$ .

# Lecture Schedule

- |                                  |             |
|----------------------------------|-------------|
| 1 Logic and Mathematical Proofs  | 6 Recursion |
| 2 Sets and Functions             | 7 Counting  |
| 3 Complexity of Algorithms       | 8 Relations |
| 4 Number Theory and Cryptography | 9 Graph     |
| 5 Mathematical Induction         | 10 Trees    |

# Recurrence

To specify a function on the basis of a recurrence:

- **Basis step (initial condition):** Specify the value of the function at zero.
- **Recursive step:** Give a rule for finding its value at an integer from its values at smaller integers.

How to formulate a recursive function?

**Find a closed-form solution?** “Top-down” and “bottom-up”

$$\begin{aligned}T(n) &= rT(n-1) + a \\&= r(rT(n-2) + a) + a \\&= r^2T(n-2) + ra + a \\&= r^2(rT(n-3) + a) + ra + a \\&= r^3T(n-3) + r^2a + ra + a \\&= r^3(rT(n-4) + a) + r^2a + ra + a \\&= r^4T(n-4) + r^3a + r^2a + ra + a.\end{aligned}\quad \begin{aligned}T(0) &= b \\T(1) &= rT(0) + a = rb + a \\T(2) &= rT(1) + a = r(rb + a) + a = r^2b + ra + a \\T(3) &= rT(2) + a = r^3b + r^2a + ra + a\end{aligned}$$

Mathematical induction.

# Lecture Schedule

- |                            |                          |
|----------------------------|--------------------------|
| 1 Logic                    | 7 Mathematical Induction |
| 2 Mathematical Proofs      | 8 Recursion              |
| 3 Sets and Functions       | 9 Counting               |
| 4 Complexity of Algorithms | 10 Relations             |
| 5 Number Theory            | 11 Graph                 |
| 6 Cryptography             | 12 Trees                 |

# Counting

**Product Rule:** If a count of elements can be broken down into a **sequence of dependent counts** where the first count yields  $n_1$  elements, the second  $n_2$  elements, and  $k$ -th count  $n_k$  elements, then the total number of elements is

$$n = n_1 \times n_2 \times \dots \times n_k$$

## Sum Rule:

- A task can be done either in one of  $n_1$  ways or in one of  $n_2$  ways
- None of the set of  $n_1$  ways is the same as any of the set of  $n_2$  ways

## The Subtraction Rule:

- A task can be done in **either  $n_1$  ways or  $n_2$  ways**
- **Principle of inclusion–exclusion:**

$$|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2|.$$



# Pigeonhole Principle

Assume that there are a set of objects and a set of bins to store them.

**The Pigeonhole Principle:** If  $k$  is a positive integer and  $k + 1$  or more objects are placed into  $k$  boxes, then there is at **least one box containing two or more** of the objects.

If  $N$  objects are placed into  $k$  bins, then there is at least one bin containing **at least  $\lceil N/k \rceil$  objects**.

**Use pigeonhole principle in proofs.**

# Permutations and Combinations

**Theorem:** If  $n$  is a positive integer and  $r$  is an integer with  $1 \leq r \leq n$ , then there are

$$P(n, r) = n(n-1)(n-2) \cdots (n-r+1)$$

$r$ -permutations of a set with  $n$  distinct elements.

**Theorem:** For integers  $n$  and  $r$  with  $0 \leq r \leq n$ , the number of  $r$ -element subsets of an  $n$ -element set is

$$\binom{n}{r} = C(n, r) = \frac{P(n, r)}{r!} = \frac{n!}{r!(n-r)!}$$



# Combinatorial Proof

**Theorem:** Let  $n$  and  $r$  be nonnegative integers with  $r \leq n$ . Then  $C(n, r) = C(n, n - r)$ .

**Definition:** A combinatorial proof of an identity is

- a proof that uses counting arguments to prove that both sides of the identity count the same objects but in different ways
- or a proof that is based on showing that there is a bijection between the sets of objects counted by the two sides of the identity.

These two types of proofs are called double counting proofs and bijective proofs, respectively.

**Use combinatorial proof to prove.**

# The Binomial Theorem

Let  $x$  and  $y$  be variables, and let  $n$  be a nonnegative integer:

$$(x + y)^n = \sum_{j=0}^n \binom{n}{j} x^{n-j} y^j = \binom{n}{0} x^n + \binom{n}{1} x^{n-1} y + \cdots + \binom{n}{n-1} x y^{n-1} + \binom{n}{n} y^n.$$

**Corollary:** Let  $n$  be a nonnegative integer,

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

**Theorem:** Let  $n$  and  $k$  be positive integers with  $n \geq k$ . Then,

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

# Generalized Permutations and Combinations

- Permutations with repetition

**Repetition:** Distinct objects; each can be selected multiple times

**Theorem:** The number of  $r$ -permutations of a set of  $n$  objects **with repetition** allowed is  $n^r$ .

- Permutations with indistinguishable objects

**Indistinguishable objects:** E.g., “SUCCESS”

**Theorem:** The number of different permutations of  $n$  objects, where there are  $n_1$  **indistinguishable objects** of type 1,  $n_2$  indistinguishable objects of type 2, . . . , and  $n_k$  indistinguishable objects of type  $k$ , is

$$C(n, n_1)(n - n_1, n_2) \cdots C(n - n_1 - \cdots n_{k-1}, n_k) = \frac{n!}{n_1!n_2! \cdots n_k!}.$$

- Combinations with repetition

# Combinations with Repetition

**Example:** How many ways are there to select five bills from a cash box containing \$1 bills, \$2 bills, \$5 bills, \$10 bills, \$20 bills, \$50 bills, and \$100 bills?

**Theorem:** There are  $C(n + r - 1, r) = C(n + r - 1, n - 1)$   $r$ -combinations from a set with  $n$  elements when repetition of elements is allowed.

# Solving Linear Homogeneous Recurrence Relations

**Definition:** A **linear homogeneous relation** of degree  $k$  with constant coefficients is a recurrence relation of the form

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \dots + c_k a_{n-k},$$

where  $c_1, c_2, \dots, c_k$  are real numbers, and  $c_k \neq 0$ .

By induction, such a recurrence relation is **uniquely** determined by this recurrence relation and  **$k$  initial conditions**  $a_0, a_1, \dots, a_{k-1}$ .

# Solving Linear Homogeneous Recurrence Relations

The characteristic equation (CE) is:

$$r^k - \sum_{i=1}^k c_i r^{k-i} = 0.$$

**Theorem:** Suppose that there are  $t$  roots  $r_1, \dots, r_t$  with multiplicities  $m_1, \dots, m_t$ . Then,

$$\begin{aligned} a_n = & (\alpha_{1,0} + \alpha_{1,1}n + \dots + \alpha_{1,m_1-1}n^{m_1-1})r_1^n \\ & + (\alpha_{2,0} + \alpha_{2,1}n + \dots + \alpha_{2,m_2-1}n^{m_2-1})r_2^n \\ & + \dots + (\alpha_{t,0} + \alpha_{t,1}n + \dots + \alpha_{t,m_t-1}n^{m_t-1})r_t^n \end{aligned}$$

- Solving the roots with CE
- Solving the  $\alpha_i$  for all  $i$  using initial conditions

# Linear Nonhomogeneous Recurrence Relations

**Definition:** A linear nonhomogeneous relation with constant coefficients may contain some terms  $F(n)$  that depend only on  $n$

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \dots + c_k a_{n-k} + F(n).$$

The recurrence relation  $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \dots + c_k a_{n-k}$  is called the associated homogeneous recurrence relation.

**Theorem:** If  $\{a_n^{(p)}\}$  is any particular solution to the linear nonhomogeneous relation with constant coefficients,

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \dots + c_k a_{n-k} + F(n),$$

Then all its solutions are of the form

$$a_n = a_n^{(p)} + a_n^{(h)},$$

where  $\{a_n^{(h)}\}$  is any solution to the associated homogeneous recurrence relation  $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \dots + c_k a_{n-k}$ .



# Linear Nonhomogeneous Recurrence Relations

Suppose that  $\{a_n\}$  satisfies the linear nonhomogeneous recurrence relation

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k} + F(n),$$

where  $c_1, c_2, \dots, c_k$  are real numbers, and

$$F(n) = (b_t n^t + b_{t-1} n^{t-1} + \cdots + b_1 n + b_0) s^n,$$

where  $b_0, b_1, \dots, b_t$  and  $s$  are real numbers. When  $s$  is not a root of the characteristic equation of the associated linear homogeneous recurrence relation, there is a particular solution of the form

$$(p_t n^t + p_{t-1} n^{t-1} + \cdots + p_1 n + p_0) s^n.$$

When  $s$  is a root of this characteristic equation and its multiplicity is  $m$ , there is a particular solution of the form

$$n^m (p_t n^t + p_{t-1} n^{t-1} + \cdots + p_1 n + p_0) s^n.$$

Pay attention to the case when  $s = 1$ .



# Generating function

The **generating function** for the sequence  $a_0, a_1, \dots, a_k, \dots$  of **real numbers** is the infinite series

$$G(x) = a_0 + a_1x + \dots + a_kx^k + \dots = \sum_{k=0}^{\infty} a_kx^k.$$

## How to use generating function to solve counting problems?

- Based on the combination problem, transfer the problem as finding the coefficient of  $x^r$  of a generating function, e.g.,

$$G(x) = (1 + x + x^2 + x^3 + \dots)^n$$

- Find the coefficient of  $x^r$ 
  - ▶ Enumerate all possibilities or
  - ▶ **Sum of geometric progression, binomial theorem**

## How to use generating function to solve linear recurrence relation?



# Lecture Schedule

- |                            |                          |
|----------------------------|--------------------------|
| 1 Logic                    | 7 Mathematical Induction |
| 2 Mathematical Proofs      | 8 Recursion              |
| 3 Sets and Functions       | 9 Counting               |
| 4 Complexity of Algorithms | 10 Relations             |
| 5 Number Theory            | 11 Graph                 |
| 6 Cryptography             | 12 Trees                 |

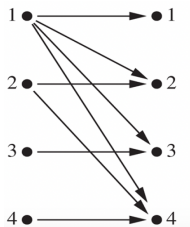
# Cartesian Product

Let  $A = \{a_1, a_2, \dots, a_m\}$  and  $B = \{b_1, b_2, \dots, b_n\}$ , the **Cartesian product**  $A \times B$  is the set of pairs  $\{(a_1, b_1), (a_2, b_2), \dots, (a_1, b_n), \dots, (a_m, b_n)\}$ .

Let  $A$  and  $B$  be two sets. A **binary relation** from  $A$  to  $B$  is a subset of a Cartesian product  $A \times B$ .

A **relation on the set  $A$**  is a relation from  $A$  to **itself**.

We use the notation  $aRb$  to denote  $(a, b) \in R$ , and  $a \not R b$  to denote  $(a, b) \notin R$ .



$R$	1	2	3	4
1	×	×	×	×
2		×		×
3			×	
4				×

# Summary on Properties of Relations

- **Reflexive Relation:** A relation  $R$  on a set  $A$  is called reflexive if  $(a, a) \in R$  for every element  $a \in A$ .
- **Irreflexive Relation:** A relation  $R$  on a set  $A$  is called irreflexive if  $(a, a) \notin R$  for every element  $a \in A$ .
- **Symmetric Relation:** A relation  $R$  on a set  $A$  is called symmetric if  $(b, a) \in R$  whenever  $(a, b) \in R$  for all  $a, b \in A$ .
- **Antisymmetric Relation:** A relation  $R$  on a set  $A$  is called antisymmetric if  $(b, a) \in R$  and  $(a, b) \in R$  implies  $a = b$  for all  $a, b \in A$ .
- **Transitive Relation:** A relation  $R$  on a set  $A$  is called transitive if  $(a, b) \in R$  and  $(b, c) \in R$  implies  $(a, c) \in R$  for all  $a, b, c \in A$ .

The number of relations?

# Composite of Relations

**Definition:** Let  $R$  be a relation from a set  $A$  to a set  $B$  and  $S$  be a relation from  $B$  to  $C$ . The composite of  $R$  and  $S$  is the relation consisting of the ordered pairs  $(a, c)$  where  $a \in A$  and  $c \in C$  and for which there is a  $b \in B$  such that  $(a, b) \in R$  and  $(b, c) \in S$ .

**Example:** Let  $A = \{1, 2, 3\}$ ,  $B = \{0, 1, 2\}$ , and  $C = \{a, b\}$ :

- $R = \{(1, 0), (1, 2), (3, 1), (3, 2)\}$
- $S = \{(0, b), (1, a), (2, b)\}$
- $S \circ R = \{(1, b), (3, a), (3, b)\}$

# Power of a Relation

**Definition:** Let  $R$  be a relation on  $A$ . The **powers**  $R^n$ , for  $n = 1, 2, 3, \dots$ , is defined inductively by

$$R^1 = R \text{ and } R^{n+1} = R^n \circ R$$

**Theorem:** The relation  $R$  on a set  $A$  is transitive if and only if  $R^n \subseteq R$  for  $n = 1, 2, 3, \dots$

How to prove?

Physical meaning of  $R^n$  in the associated directed graph?

# Representing Relations

Some special ways to represent **binary relations**:

- with a zero-one matrix
- with a directed graph

$$\begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 & \\ & & & & & 1 \end{bmatrix}$$

Reflexive

$$\begin{bmatrix} & 1 & \\ 1 & & \\ & & 0 \end{bmatrix}$$

Symmetric

$$\begin{bmatrix} & 1 & 0 \\ 0 & & \\ 0 & 1 & \end{bmatrix}$$

Antisymmetric

# Zero-One Matrix

Consider relations  $R_1$  and  $R_2$  on a set  $A$ :

$$M_{R_1 \cup R_2} = M_{R_1} \vee M_{R_2}$$

$$M_{R_1 \cap R_2} = M_{R_1} \wedge M_{R_2}$$

Suppose that  $R$  is a relation from  $A$  to  $B$  and  $S$  is a relation from  $B$  to  $C$ :

$$M_{S \circ R} = M_R \odot M_S.$$

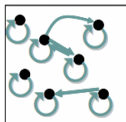
The ordered pair  $(a_i, c_j)$  belongs to  $S \circ R$  **if and only if** there is an element  $b_k$  such that  $(a_i, b_k)$  belongs to  $R$  and  $(b_k, c_j)$  belongs to  $S$ .



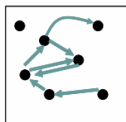
# Directed Graph

A **directed graph**, or digraph, consists of a set  $V$  of **vertices** together with a set  $E$  of ordered pairs of elements of  $V$  called **edges**.

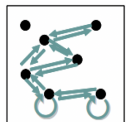
The vertex  $a$  is called the **initial vertex** of the edge  $(a, b)$ , and the vertex  $b$  is called the **terminal vertex** of this edge.



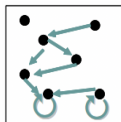
reflexive



irreflexive



symmetric



antisymmetric



SUSTech

Southern University  
of Science and  
Technology

# Closures of Relations

The set  $S$  is called the **reflexive closure** of  $R$  if it:

- contains  $R$
- is reflexive
- is minimal (is contained in **every** reflexive relation  $Q$  that contains  $R$  ( $R \subseteq Q$ ), i.e.,  $S \subseteq Q$ )

## Proof?

Relations can have different **properties**:

- Reflexive closures
- Symmetric closures
- Transitive closures: Finding a **transitive closure** corresponds to finding all pairs of elements that are connected with a directed path.

# Paths in Directed Graphs

**Definition:** A path from  $a$  to  $b$  in the directed graph  $G$  is a sequence of edges  $(x_0, x_1), (x_1, x_2), \dots, (x_{n-1}, x_n)$  in  $G$ , where  $n$  is nonnegative and  $x_0 = a$  and  $x_n = b$ .

A path of length  $n \geq 1$  that begins and ends at the same vertex is called a circuit or cycle.

**Theorem:** Let  $R$  be relation on a set  $A$ . There is a path of length  $n$  from  $a$  to  $b$  if and only if  $(a, b) \in R^n$ .

**Proof** (by induction)

# Connectivity Relation

**Definition:** Let  $R$  be a relation on a set  $A$ . The **connectivity relation**  $R^*$  consists of **all pairs**  $(a, b)$  such that there is a path (of any length) between  $a$  and  $b$  in  $R$ :

$$R^* = \bigcup_{k=1}^{\infty} R^k$$

**Theorem:** The transitive closure of a relation  $R$  equals the connectivity relation  $R^*$ .

**Prove**  $R^*$  **is a transitive closure of**  $R$ ?

# Find Transitive Closure

**Lemma:** If there is a path of length at least one in  $R$  from  $a$  to  $b$ , then there is such a path with length **not exceeding**  $n$ .

Thus,

$$R^* = R \cup R^2 \cup R^3 \cup \dots \cup R^n.$$

**Theorem:** Let  $M_R$  be the zero-one matrix of the relation  $R$  on a set with  $n$  elements. Then the zero-one matrix of the transitive closure  $R^*$  is

$$M_{R^*} = M_R \vee M_R^{[2]} \vee M_R^{[3]} \vee \dots \vee M_R^{[n]},$$

where  $M_R^{[n]} = \underbrace{M_R \odot M_R \odot \dots \odot M_R}_{n \text{ } M'_R \text{'s}}$

# Roy-Warshall Algorithm

Consider a list of vertices  $v_1, v_2, \dots, v_k, \dots, v_n$ . Define a zero-one matrix

$$\mathbf{W}_k = [w_{ij}^{(k)}],$$

where  $w_{ij}^{(k)} = 1$  if there is a path from  $v_i$  to  $v_j$  such that **all the interior vertices** of this path are in the set  $\{v_1, v_2, \dots, v_k\}$  and is 0 otherwise.

Warshall's algorithm computes  $M_{R^*}$  by efficiently computing

$$\mathbf{W}_0 = M_R, W_1, W_2, \dots, \mathbf{W}_n = M_{R^*}.$$

## ALGORITHM 2 Warshall Algorithm.

**procedure** *Warshall* ( $\mathbf{M}_R : n \times n$  zero-one matrix)

$\mathbf{W} := \mathbf{M}_R$

**for**  $k := 1$  **to**  $n$

**for**  $i := 1$  **to**  $n$

**for**  $j := 1$  **to**  $n$

$w_{ij} := w_{ij} \vee (w_{ik} \wedge w_{kj})$

**return**  $\mathbf{W}$  { $\mathbf{W} = [w_{ij}]$  is  $\mathbf{M}_{R^*}$ }

# Equivalence Relation

**Definition:** A relation  $R$  on a set  $A$  is called an **equivalence relation** if it is reflexive, symmetric, and transitive.

**Definition:** Let  $R$  be an **equivalence relation on a set  $A$** . The set of all elements that are related to an **element  $a$**  of  $A$  is called **the equivalence class of  $a$** , denoted by  $[a]_R$ . When only one relation is considered, we use the notation  $[a]$ .

$$[a]_R = \{b : (a, b) \in R\}$$

**Theorem:** Let  $R$  be an **equivalence relation** on a set  $A$ . The following statements are equivalent:

$$(i) aRb \quad (ii) [a] = [b] \quad (iii) [a] \cap [b] \neq \emptyset$$

# Partition of a Set $S$

**Definition:** Let  $S$  be a set. A collection of nonempty subsets of  $S$ , i.e  $A_1, A_2, \dots, A_k$ , is called a partition of  $S$  if:

$$A_i \cap A_j = \emptyset, i \neq j \text{ and } S = \bigcup_{i=1}^k A_i$$

**Theorem:** The equivalence classes form a partition of  $A$ .

**Theorem:** Let  $\{A_1, A_2, \dots, A_i, \dots\}$  be a partition of  $S$ . Then, there is an equivalence relation  $R$  on  $S$ , that has the sets  $A_i$  as its equivalence classes.



# Partial Ordering

**Definition:** A relation  $R$  on a set  $S$  is called a **partial ordering**, or partial order, if it is **reflexive**, **antisymmetric**, and **transitive**.

A set  $S$  together with a partial ordering  $R$  is called a **partially ordered set**, or **poset**, denoted by  $(S, R)$ .

The notation  $a \preceq b$  is used to denote that  $(a, b) \in R$  in an arbitrary poset  $(S, R)$ .

The notation  $a \prec b$  denotes that  $a \preceq b$ , but  $a \neq b$ .

# Comparability

**Definition:** The elements  $a$  and  $b$  of a poset  $(S, \preceq)$  are comparable if either  $a \preceq b$  or  $b \preceq a$ . Otherwise,  $a$  and  $b$  are called incomparable.

**Definition:** If  $(S, \preceq)$  is a poset and every two elements of  $S$  are comparable,  $S$  is called a totally ordered or linearly ordered set, and  $\preceq$  is called a total order or a linear order.

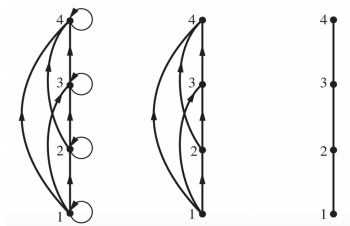
$(S, \preceq)$  is a well-ordered set if it is a poset such that  $\preceq$  is a total ordering and every nonempty subset of  $S$  has a least element.

## The Principle of Well-Ordered Induction

# Hasse Diagram

Start with the directed graph of the relation:

- Remove the loops  $(a, a)$  present at every vertex due to the reflexive property.
- Remove all edges  $(x, y)$  for which there is an element  $z \in S$  s.t.  $x \prec z$  and  $z \prec y$ . These are the edges that must be present due to the transitive property.
- Arrange each edge so that its initial vertex is below the terminal vertex. Remove all the arrows, because all edges point upwards toward their terminal vertex.



# Maximal and Minimal Elements

**Definition:**  $a$  is a **maximal** (resp. **minimal**) element in poset  $(S, \preceq)$  if there is no  $b \in S$  such that  $a \prec b$  (resp.  $b \prec a$ ).

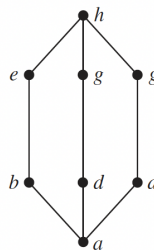
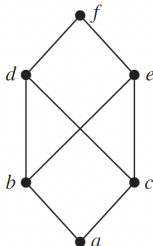
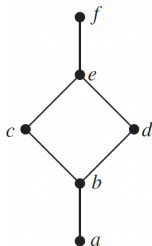
**Definition:**  $a$  is the **greatest** (resp. **least**) element of the poset  $(S, \preceq)$  if  $b \preceq a$  (resp.  $a \preceq b$ ) **for all**  $b \in S$ .

**Definition:** Let  $A$  be a subset of a poset  $(S, \preceq)$ .

- $u \in S$  is called an **upper bound** (resp. lower bound) of  $A$  if  $a \preceq u$  (resp.  $u \preceq a$ ) **for all**  $a \in A$ .
- $x \in S$  is called the **least upper bound** (resp. greatest lower bound) of  $A$  if  $x$  is an upper bound (resp. lower bound) that is **less than any other** upper bounds (resp. lower bounds) of  $A$ .

# Upper and Lower Bound

**Definition:** A partial ordered set in which **every pair of elements** has **both** a least upper bound and a greatest lower bound is called a **lattice**.

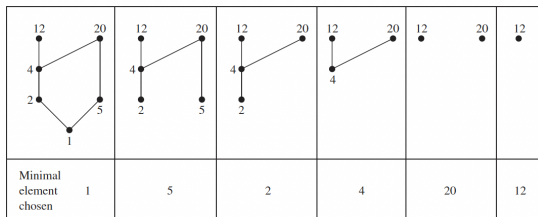


- (a) and (c): lattices
- (b): **not a lattice**, because the elements  $b$  and  $c$  have **no least upper bound**.

# Topological Sorting for Finite Posets

**Topological sorting:** Given a partial ordering  $R$ , find a total ordering  $\preceq$  such that  $a \preceq b$  whenever  $aRb$ .  $\preceq$  is said compatible with  $R$ .

Find a compatible total ordering for the poset  $(\{1, 2, 4, 5, 12, 20\}, |)$ .



This produces the total ordering

$$1 \prec 5 \prec 2 \prec 4 \prec 20 \prec 12$$

# Lecture Schedule

- |                            |                          |
|----------------------------|--------------------------|
| 1 Logic                    | 7 Mathematical Induction |
| 2 Mathematical Proofs      | 8 Recursion              |
| 3 Sets and Functions       | 9 Counting               |
| 4 Complexity of Algorithms | 10 Relations             |
| 5 Number Theory            | 11 Graph                 |
| 6 Cryptography             | 12 Trees                 |

# Definition of a Graph

A **graph**  $G = (V, E)$  consists of a nonempty set  $V$  of vertices (or nodes) and a set  $E$  of edges. Each edge has either one or two vertices associated with it, called its endpoints. An edge is said to be incident to (or connect) its endpoints.

**Simple graph**: A graph in which each edge connects two **different** vertices and where **no** two edges connect the same pair of vertices.

Two vertices  $u, v$  in an **undirected graph**  $G$  are called **adjacent** (or neighbors) in  $G$  if there is an edge  $e$  between  $u$  and  $v$ . Such an edge  $e$  is called **incident** with the vertices  $u$  and  $v$  and  $e$  is said to connect  $u$  and  $v$ .



# Undirected Graphs

**Definition:** The set of all neighbors of a vertex  $v$  of  $G = (V, E)$ , denoted by  $N(v)$ , is called the **neighborhood of  $v$** .

If  $A$  is a **subset** of  $V$ , we denote by  $N(A)$  the set of all vertices in  $G$  that are adjacent to **at least one** vertex in  $A$ .

**Definition:** The degree of a vertex in an undirected graph is the **number of edges incident with it**, except that a **loop** at a vertex contributes **two** to the degree of that vertex. The degree of the vertex  $v$  is denoted by  $\deg(v)$ .

**Theorem** (Handshaking Theorem): If  $G = (V, E)$  is an **undirected** graph with  $m$  edges, then

$$2m = \sum_{v \in V} \deg(v)$$

# Directed Graphs

**Definition:** Let  $(u, v)$  be an edge in  $G$ . Then  $u$  is the **initial vertex** of the edge and is **adjacent to  $v$**  and  $v$  is the **terminal vertex** of this edge and is **adjacent from  $u$** . The initial and terminal vertices of a loop are the same.

**Definition:** The **in-degree** of a vertex  $v$ , denoted by  $\deg^-(v)$ , is the number of edges which terminate at  $v$ . The **out-degree** of  $v$ , denoted by  $\deg^+(v)$ , is the number of edges with  $v$  as their initial vertex.

Note that a loop at a vertex contributes 1 to both the in-degree and the out-degree of the vertex.

**Theorem:** Let  $G = (V, E)$  be a graph with directed edges. Then,

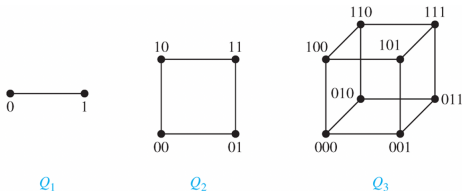
$$|E| = \sum_{v \in V} \deg^-(v) = \sum_{v \in V} \deg^+(v)$$

# Special Graphs

A **complete graph** on  $n$  vertices, denoted by  $K_n$ , is the simple graph that contains exactly one edge between each pair of distinct vertices.

A **cycle**  $C_n$  for  $n \geq 3$  consists of  $n$  vertices  $v_1, v_2, \dots, v_n$ , and edges  $\{v_1, v_2\}, \{v_2, v_3\}, \dots, \{v_{n-1}, v_n\}, \{v_n, v_1\}$ .

An  **$n$ -dimensional hypercube**, or  $n$ -cube,  $Q_n$  is a graph with  $2^n$  vertices representing all bit strings of length  $n$ , where there is an edge between two vertices that **differ in exactly one bit position**.



# Bipartite Graphs

**Definition:** A simple graph  $G$  is **bipartite** if  $V$  can be partitioned into two disjoint subsets  $V_1$  and  $V_2$  such that **every edge** connects a vertex in  $V_1$  and a vertex in  $V_2$ .

**Definition:** A **complete bipartite graph**  $K_{m,n}$  is a graph that has its vertex set partitioned into two subsets  $V_1$  of size  $m$  and  $V_2$  of size  $n$  such that there is an edge from **every** vertex in  $V_1$  to **every** vertex in  $V_2$ .

# Bipartite Graphs and Matchings

**Matching** the elements of one set to elements in another. A matching is a **subset of edges** such that no two edges are incident with the same vertex.

A matching  $M$  in a bipartite graph  $G = (V, E)$  with bipartition  $(V_1, V_2)$  is a **complete matching** from  $V_1$  to  $V_2$  if every vertex in  $V_1$  is the endpoint of an edge in the matching, or equivalently, if  $|M| = |V_1|$ .

**Theorem** (Hall's Marriage Theorem): The bipartite graph  $G = (V, E)$  with bipartition  $(V_1, V_2)$  has a complete matching from  $V_1$  to  $V_2$  if and only if  $|N(A)| \geq |A|$  for all subsets  $A$  of  $V_1$ .

**Proof?**

# Subgraphs

**Definition:** A **subgraph of a graph**  $G = (V, E)$  is a graph  $(W, F)$ , where  $W \subseteq V$  and  $F \subseteq E$ .

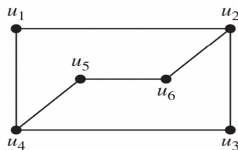
**Definition:** The **union of two simple graphs**  $G_1 = (V_1, E_1)$  and  $G_2 = (V_2, E_2)$  is the simple graph with vertex set  $V_1 \cup V_2$  and edge set  $E_1 \cup E_2$ , denoted by  $G_1 \cup G_2$ .

To represent a graph, we may use **adjacency lists**, **adjacency matrices**, and **incidence matrices**.

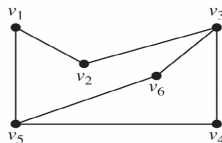
# Isomorphism of Graphs

**Definition:** The simple graphs  $G_1 = (V_1, E_1)$  and  $G_2 = (V_2, E_2)$  are **isomorphic** if there is a **one-to-one and onto function** from  $V_1$  to  $V_2$  with the property that  $a$  and  $b$  are adjacent in  $G_1$  if and only if  $f(a)$  and  $f(b)$  are adjacent in  $G_2$ , for all  $a$  and  $b$  in  $V_1$ . Such a function is called an **isomorphism**.

Useful graph invariants include the **number of vertices**, **number of edges**, **degree sequence**, existence circuit with certain length, etc.



$G$



$H$



SUSTech

Southern University  
of Science and  
Technology

## Path: Undirected Graph

**Definition:** Let  $n$  be a nonnegative integer and  $G$  an **undirected** graph. A **path of length  $n$  from  $u$  to  $v$**  in  $G$  is a sequence of  $n$  edges  $e_1, e_2, \dots, e_n$  of  $G$  for which there exists a sequence  $x_0 = u, x_1, \dots, x_{n-1}, x_n = v$  of vertices such that  $e_i$  has the endpoints  $x_{i-1}$  and  $x_i$  for  $i = 1, \dots, n$ .

The path is a **circuit** if it begins and ends at the same vertex, i.e., if  $u = v$ , and has length greater than zero.

A path or circuit is **simple** if it does not contain **the same edge** more than once.

Length of a path = the number of edges on path

Path and circuit in directed graph?



# Connectivity

An undirected graph is called **connected** if there is a path between **every pair** of distinct vertices of the graph.

**Lemma:** If there is a path between two distinct vertices  $x$  and  $y$  of a graph  $G$ , then there is a simple path between  $x$  and  $y$  in  $G$ .

**Theorem:** There is a **simple path** between every pair of distinct vertices of a **connected** undirected graph.

# Connectivity

A **connected component** of a graph  $G$  is a **connected** subgraph of  $G$  that is **not a proper** subgraph of another connected subgraph of  $G$ .

A graph  $G$  that is not connected has two or more connected components that are disjoint and have  $G$  as their union.

**Definition:** A directed graph is **strongly connected** if there is a path from  $a$  to  $b$  **and** a path from  $b$  to  $a$  whenever  $a$  and  $b$  are vertices in the graph.

**Definition:** A directed graph is **weakly connected** if there is a path between every two vertices in the **underlying undirected graph**.

# Cut Vertices and Cut Edges

Sometimes the **removal** from a graph of a vertex and all incident edges disconnect the graph. Such vertices are called **cut vertices**.

Similarly, we define **cut edges**.

A set of edges  $E'$  is called an edge cut of  $G$  if the subgraph  $G - E'$  is disconnected. The **edge connectivity**  $\lambda(G)$  is the **minimum** number of edges in an edge cut of  $G$ .

# Counting Paths between Vertices

**Theorem:** Let  $G$  be a graph with adjacency matrix  $\mathbf{A}$  with respect to the ordering  $v_1, v_2, \dots, v_n$  of vertices. The number of different paths of length  $r$  from  $v_i$  to  $v_j$ , where  $r$  is a positive integer, equals the  $(i, j)$ -th entry of  $\mathbf{A}^r$ .

**Proof (by induction)**

Note: with directed or undirected edges, multiple edges and loops allowed

# Euler Paths

**Definition:** An **Euler circuit** in a graph  $G$  is a **simple circuit** containing every edge of  $G$ . An Euler path in  $G$  is a simple path containing every edge of  $G$ .

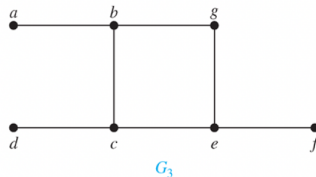
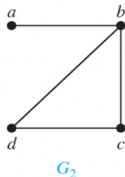
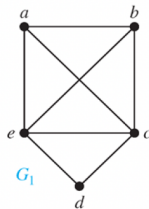
**Theorem:** A connected multigraph with at least two vertices has an **Euler circuit** if and only if each of its vertices has **even degree**.

**Theorem:** A connected multigraph has an Euler path but not an **Euler circuit** if and only if it has exactly **two vertices of odd degree**.

# Hamilton Paths and Circuits

**Definition:** A simple path in a graph  $G$  that passes through every vertex exactly once is called a **Hamilton path**, and a simple circuit in a graph  $G$  that passes through every vertex exactly once is called a **Hamilton circuit**.

**Example:** Which of these simple graphs has a Hamilton circuit or, if not, a Hamilton path?



- $G_1$  has a Hamilton circuit:  $a, b, c, d, e, a$ ;
- $G_2$  has no Hamilton circuit (because containing every vertex must contain the edge  $a, b$  twice), but it has a Hamilton path;
- $G_3$  has neither, because any path containing all vertices must contain one of the edges  $\{a, b\}$ ,  $\{e, f\}$ , and  $\{c, d\}$  more than once.

# Planar Graphs

**Definition:** A graph is called **planar** if it can be drawn in the **plane without any edges crossing**. Such a drawing is called a **planar representation** of the graph.

A planar representation of a graph splits the plane into **regions**, including an unbounded region.

**Theorem (Euler's Formula):** Let  $G$  be a **connected planar simple graph** with  $e$  edges and  $v$  vertices. Let  $r$  be the number of regions in a planar representation of  $G$ . Then,  $r = e - v + 2$ .

**Proof?**

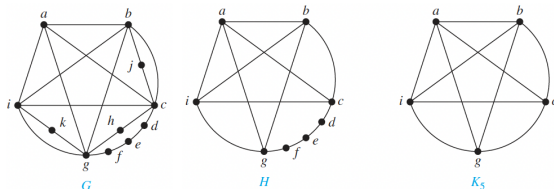
# Kuratowski's Theorem

If a graph is planar, **so will be any graph** obtained by removing an edge  $\{u, v\}$  and adding a new vertex  $w$  together with edges  $\{u, w\}$  and  $\{w, v\}$ . Such an operation is called an **elementary subdivision**.

The graphs  $G_1 = (V_1, E_1)$  and  $G_2 = (V_2, E_2)$  are called **homeomorphic** if they can be obtained from the same graph by a sequence of elementary subdivisions.

**Theorem:** A graph is **nonplanar** if and only if it contains a **subgraph homomorphic** to  $K_{3,3}$  or  $K_5$ .

**Example:**





# Lecture Schedule

- |                            |                          |
|----------------------------|--------------------------|
| 1 Logic                    | 7 Mathematical Induction |
| 2 Mathematical Proofs      | 8 Recursion              |
| 3 Sets and Functions       | 9 Counting               |
| 4 Complexity of Algorithms | 10 Relations             |
| 5 Number Theory            | 11 Graph                 |
| 6 Cryptography             | 12 <del>Trees</del>      |