离散数学课堂笔记/ Discrete Mathematics Notes

参考资料 / Main references:
MIT 6.042J / 6.1200J *Mathematics for Computer Science*, Rosen *Discrete Mathematics and Its Applications*,《离散数学及其应用》中文版，以及本课程课件、quiz 与练习材料。

1. 逻辑与证明 / Logic and Proofs

1.1 命题与联结词 / Propositions and Connectives

Definition / 定义

命题是具有确定真值的陈述。A proposition is a declarative sentence that is either true or false.

常用联结词 / Connectives:

- 非 / negation: $\neg p$，当且仅当 p 为假时为真。
- 且 / conjunction: $p \wedge q$，当且仅当 p 和 q 都真时为真。
- 或 / disjunction: $p \vee q$，当至少一个为真时为真。
- 异或 / exclusive or: $p \oplus q$，当恰好一个为真时为真。
- 蕴含 / implication: $p \to q$，只有 p 真且 q 假时为假。
- 双条件 / biconditional: $p \leftrightarrow q$，当 p 与 q 真值相同为真。

Key formula / 核心公式

$$\begin{aligned} p \to q &\equiv \neg p \vee q, \\ p \leftrightarrow q &\equiv (p \to q) \wedge (q \to p), \\ p \oplus q &\equiv (p \vee q) \wedge \neg(p \wedge q). \end{aligned} \tag{1}$$

Proof / 证明

用真值表验证。对 p 和 q 的每一种真假赋值，两边真值都相同。以蕴含为例，$p \to q$ 和 $\neg p \vee q$ 都只在 $p = T, q = F$ 时为假。

Use / 怎么用

遇到复杂逻辑式，先把 $\to$ 与 $\leftrightarrow$ 改成 $\neg, \wedge, \vee$，再用德摩根律和分配律化简。In proofs, replacing implication by $\neg p \vee q$ is often the fastest way to convert to CNF or DNF.

Pitfall / 易错点

$p \to q$ 不是“因果关系”，只是一个真值函数。当前件 p 为假时，整个蕴含为真。

深讲：真值表、永真式、矛盾式 / Truth Tables, Tautologies, Contradictions

Definitions / 定义

一个复合命题如果在所有真值赋值下都为真，称为永真式 / tautology。如果在所有赋值下都为假，称为矛盾式 / contradiction。如果有时真有时假，称为 contingency / 偶然式（可满足但非永真）。

How truth tables work / 真值表为什么可靠

含有 n 个不同命题变量的公式，真值只由这些变量的真假决定。每个变量有 2 种取值，所以总共有 2^n 行。若两个公式在全部 2^n 行中真值相同，它们就逻辑等价。

Example / 例题：证明吸收律

证明:

$$p \vee (p \wedge q) \equiv p. \tag{2}$$

不用完整真值表，也可以用语义解释。若 p 为真，则左边 $p \vee (p \wedge q)$ 为真，右边也为真。若 p 为假，则 $p \wedge q$ 为假，所以左边为假，右边也为假。因此两边等价。

Algebraic proof / 代数证明

$$\begin{aligned} p \vee (p \wedge q) &\equiv (p \wedge T) \vee (p \wedge q) \\ &\equiv p \wedge (T \vee q) \\ &\equiv p \wedge T \\ &\equiv p. \end{aligned} \tag{3}$$

How to use / 怎么用

吸收律常用于化简逻辑式和集合式。对应集合恒等式是：

$$A \cup (A \cap B) = A, \qquad A \cap (A \cup B) = A. \tag{4}$$

Trap / 陷阱

真值表证明最稳，但行数随变量数指数增长。变量多时优先使用等价律或结构化证明。

1.2 逻辑等价律 / Logical Equivalences

Theorem / 定理

以下等价式在任意命题 p, q, r 下成立。

$$\begin{aligned} p \wedge T &\equiv p, & p \vee F &\equiv p, \\ p \vee T &\equiv T, & p \wedge F &\equiv F, \\ p \vee p &\equiv p, & p \wedge p &\equiv p, \\ \neg\neg p &\equiv p, & & \\ p \vee q &\equiv q \vee p, & p \wedge q &\equiv q \wedge p, \\ (p \vee q) \vee r &\equiv p \vee (q \vee r), & (p \wedge q) \wedge r &\equiv p \wedge (q \wedge r), \\ p \vee (q \wedge r) &\equiv (p \vee q) \wedge (p \vee r), & & \\ p \wedge (q \vee r) &\equiv (p \wedge q) \vee (p \wedge r), & & \\ \neg(p \wedge q) &\equiv \neg p \vee \neg q, & & \\ \neg(p \vee q) &\equiv \neg p \wedge \neg q, & & \\ p \vee (p \wedge q) &\equiv p, & p \wedge (p \vee q) &\equiv p, \\ p \vee \neg p &\equiv T, & p \wedge \neg p &\equiv F. \end{aligned} \tag{5}$$

Proof / 证明

Rosen 的标准证明方法是真值表。MIT 更强调把它们看成布尔代数规则。每条等价式两边对所有真值赋值一致，因此等价。

Use / 怎么用

- 化简命题公式：优先用 $p \to q \equiv \neg p \vee q$，再用德摩根律把否定推进到原子命题。
- 证明两个逻辑式等价：可以用真值表，也可以从一边开始逐步使用等价律。
- 写程序条件判断：德摩根律常用于把复杂条件取反。

Pitfall / 易错点

分配律有两条，逻辑中 $\wedge$ 对 $\vee$ 分配，$\vee$ 也对 $\wedge$ 分配；不要只记代数里乘法对加法分配的版本。

课堂展开：逻辑等价与证明写作 / Equivalences and Proof Writing

Worked proof / 例题：证明假言三段论

证明：若 $p \to q$ 且 $q \to r$，则 $p \to r$。

Direct semantic proof / 语义证明

假设 $p \to q$ 和 $q \to r$ 都为真。要证明 $p \to r$，只需考虑 p 为真的情况。若 p 真，由 $p \to q$ 得 q 真；由 $q \to r$ 得 r 真。所以 p 真时 r 真，因此 $p \to r$ 为真。

Algebraic proof / 等价式证明

目标公式为：

$$((p \to q) \land (q \to r)) \to (p \to r). \tag{6}$$

把蕴含改写：

$$(p \to q) \land (q \to r) \equiv (\neg p \lor q) \land (\neg q \lor r). \tag{7}$$

如果前件为真且 p 真，那么第一项强迫 q 真，第二项强迫 r 真。所以不可能出现前件真、p 真、r 假的反例。

How to use / 怎么用

写证明时不一定要列完整真值表。若题目是“用推理规则证明”，就写假设与推出；若题目是“证明永真式”，真值表、等价变形或反例排除都可以。

Common mistake / 常见错误

从 $p \to q$ 和 q 不能推出 p。这叫 affirming the consequent / 肯定后件谬误。

1.3 谓词与量词 / Predicates and Quantifiers

Definition / 定义

谓词 $P(x)$ 是含变量的命题函数。量词把变量绑定成命题。

- 全称量词 / universal quantifier: $\forall x P(x)$ means $P(x)$ is true for every x.
- 存在量词 / existential quantifier: $\exists x P(x)$ means there is at least one x with $P(x)$ true.
- 唯一存在 / unique existence: $\exists! x P(x)$ means exactly one x satisfies $P(x)$.

Theorem / 量词否定律

$$\begin{aligned} \neg \forall x\, P(x) &\equiv \exists x\, \neg P(x), \\ \neg \exists x\, P(x) &\equiv \forall x\, \neg P(x), \\ \exists! x\, P(x) &\equiv \exists x \Big(P(x) \land \forall y\, (P(y) \to y = x) \Big). \end{aligned} \tag{8}$$

Proof / 证明

“并非所有 x 都满足 P” 等价于“至少有一个 x 不满足 P”。“不存在满足 P 的 x” 等价于“每个 x 都不满足 P”。唯一存在分成两部分：存在一个见证 x，并且任意另一个满足 P 的 y 都等于它。

Use / 怎么用

- 反驳全称命题：找一个反例，即证明 $\exists x\, \neg P(x)$。
- 证明存在命题：构造一个见证 a 并验证 $P(a)$。
- 证明唯一性：先证明存在，再假设 a 和 b 都满足条件，推出 $a = b$。

Pitfall / 易错点

量词顺序通常不能交换：

$$\forall x\, \exists y\, P(x, y) \not\equiv \exists y\, \forall x\, P(x, y). \tag{9}$$

前者允许 y 随 x 改变；后者要求同一个 y 对所有 x 都有效。

课堂展开：逻辑：从中文句子到符号 / Logic: Translating Statements

核心目标 / Goal

逻辑题最重要的不是背符号，而是会把一句话翻译成可以操作的形式。Translation is the bridge between natural language and proof.

常见关键词 / Keywords

- “所有”“任意”“每一个”通常翻译为 $\forall$。
- “存在”“至少一个”“有某个”通常翻译为 $\exists$。
- “至多一个”翻译为任意两个满足条件的对象相等。
- “恰好一个”翻译为存在且唯一。
- “若……则……”翻译为 $p \to q$。
- “只有当”容易反向：“p 只有当 q”的意思是“只要 p 成立，q 必须成立”，即 $p \to q$。

Example 1 / 例题 1

把“每个整数都有一个比它大的整数”翻译成逻辑式。

Solution / 解法

对象是整数，所以变量范围是 $\mathbb{Z}$。句子中的“每个整数”是外层全称量词，“有一个比它大的整数”是内层存在量词：

$$\forall n \in \mathbb{Z}\, \exists m \in \mathbb{Z}\, (m > n). \tag{11}$$

这个命题是真的，因为给定任意 n，可取见证 $m = n + 1$。

How to use / 怎么用

证明带 $\forall n \exists m$ 的命题时，先取任意 n，再根据 n 构造 m。这里 m 可以依赖 n。

Example 2 / 例题 2

把“存在一个整数比所有整数都大”翻译成逻辑式，并判断真假。

Solution / 解法

$$\exists m \in \mathbb{Z}\, \forall n \in \mathbb{Z}\, (m > n). \tag{12}$$

这是假的。若存在这样的 m，取 $n = m + 1$，则应有 $m > m + 1$，矛盾。

Key point / 关键点

$$\forall n\, \exists m\, (m > n) \quad \text{and} \quad \exists m\, \forall n\, (m > n) \tag{13}$$

完全不同。前者的 m 可随 n 改变；后者要求同一个 m 同时大于所有整数。

Nested negation / 嵌套量词取反

要否定

$$\forall x\, \exists y\, \forall z\, P(x, y, z), \tag{14}$$

逐层翻转量词并把否定推到谓词：

$$\exists x\, \forall y\, \exists z\, \neg P(x, y, z). \tag{15}$$

Use / 怎么用

反驳复杂命题时不要凭直觉写反义句，按规则机械操作最安全：$\forall$ 和 $\exists$ 互换，最后否定谓词。

1.4 推理规则 / Rules of Inference

Core rules / 核心推理规则

- Modus ponens / 肯定前件: from p and $p \to q$, infer q.
- Modus tollens / 否定后件: from $\neg q$ and $p \to q$, infer $\neg p$.
- Hypothetical syllogism / 假言三段论: from $p \to q$ and $q \to r$, infer $p \to r$.
- Disjunctive syllogism / 析取三段论: from $p \lor q$ and $\neg p$, infer q.
- Resolution / 归结: from $p \lor q$ and $\neg p \lor r$, infer $q \lor r$.
- Universal instantiation / 全称实例化: from $\forall x P(x)$, infer $P(c)$.
- Existential instantiation / 存在实例化: from $\exists x P(x)$, choose a new witness c with $P(c)$.
- Universal generalization / 全称推广: if $P(c)$ is proved for arbitrary c, infer $\forall x P(x)$.
- Existential generalization / 存在推广: from $P(c)$, infer $\exists x P(x)$.

Proof / 证明

每条规则都可由真值表或量词语义证明。例如 modus tollens 来自逆否等价：$p \to q \equiv \neg q \to \neg p$。

Use / 怎么用

在形式证明中，每一步都注明使用了哪条规则。In English proof writing, these rules justify why a conclusion follows from earlier lines.

1.5 证明方法 / Proof Methods

1.5.1 直接证明 / Direct Proof

Pattern / 模板

要证明 $p \to q$，假设 p 为真，通过定义、已知定理和代数变形推出 q。

Example theorem / 示例定理

如果 n 是偶数，则 n^2 是偶数。

Proof / 证明

若 n 为偶数，则存在整数 k 使 $n = 2k$。于是：

$$n^2 = (2k)^2 = 4k^2 = 2(2k^2). \tag{16}$$

因为 $2k^2$ 是整数，所以 n^2 是偶数。

Use / 怎么用

题目给出清楚条件时优先直接证明，例如整除、集合包含、函数性质。

1.5.2 逆否证明 / Proof by Contrapositive

Theorem / 定理

$$p \to q \equiv \neg q \to \neg p. \tag{17}$$

Proof / 证明

两者都只在 p 真且 q 假时为假，因此真值表相同。

Use / 怎么用

当结论 q 很难直接推出，但 $\neg q$ 能强迫 $\neg p$ 时使用。典型题：证明“如果 n^2 是偶数，则 n 是偶数”。逆否为“如果 n 是奇数，则 n^2 是奇数”。

1.5.3 反证法 / Proof by Contradiction

Pattern / 模板

要证明命题 p，假设 $\neg p$，推出矛盾 F。

$$\neg p \to F \quad \Longrightarrow \quad p. \tag{18}$$

Example theorem / 示例定理

$\sqrt{2}$ 是无理数。

Proof / 证明

假设 $\sqrt{2} = a/b$，其中 a, b 互素且 $b \neq 0$。则 $a^2 = 2b^2$，所以 a^2 偶，进而 a 偶。令 $a = 2c$，得 $4c^2 = 2b^2$，所以 $b^2 = 2c^2$，从而 b 也偶。这与 a, b 互素矛盾。

Use / 怎么用

常用于无理性、无限性、不可数性、图论不可能性。

1.5.4 分类讨论 / Proof by Cases

Pattern / 模板

若 $p_1 \vee p_2 \vee \cdots \vee p_k$ 覆盖所有情况，并且每种情况都推出 q，则 q 成立。

$$(p_1 \vee \cdots \vee p_k) \wedge \bigwedge_{i=1}^{k} (p_i \to q) \quad \Longrightarrow \quad q. \tag{19}$$

Use / 怎么用

常见分类：奇偶、正负、是否为零、图中顶点度数、递推初值范围。

深讲：证明方法选择表 / Choosing a Proof Method

Direct proof / 直接证明适合什么

当假设能直接展开定义时用直接证明。例如证明 $a \mid b$ 与 $a \mid c$ 推出 $a \mid (b + c)$，直接写 $b = ak$、$c = a\ell$ 即可。

Contrapositive / 逆否适合什么

当结论是否定形式，或原命题很难从前件推出后件时，用逆否。例：证明若 n^2 为偶数，则 n 为偶数。逆否是若 n 为奇数，则 n^2 为奇数。

Contradiction / 反证适合什么

当要证明“不存在”“无限多”“无理数”“不可数”时，反证通常自然。反证的关键是把否定命题写清楚。

Induction / 归纳适合什么

命题含有自然数参数 n，并且 $n + 1$ 情况能由 n 或更小情况推出。求和公式、递推式、树的边数、算法正确性都常用归纳。

Cases / 分类讨论适合什么

对象天然分成有限情况：奇偶、正负、是否为零、图中顶点度数、集合是否为空。

How to write elegantly / 英文证明常用句式

- Let x be arbitrary. / 任取 x。
- Suppose, for contradiction, that ... / 反设……。
- By the induction hypothesis, ... / 由归纳假设……。
- Hence the two sets are equal by double inclusion. / 由双向包含，两个集合相等。
- This contradicts ..., so the assumption was false. / 这与……矛盾，所以假设错误。

2. 集合、函数、序列、求和、矩阵 / Sets, Functions, Sequences, Sums, Matrices

2.1 集合基础 / Set Basics

Definition / 定义

集合是对象的无序聚集。$x \in A$ 表示 x 属于 A。$A \subseteq B$ 表示任意 $x \in A$ 都有 $x \in B$。

Theorem / 集合相等证明原则

$$A = B \iff (A \subseteq B) \land (B \subseteq A). \tag{20}$$

Proof / 证明

若 $A = B$，元素完全相同，所以互相包含。反过来，若互相包含，则任意元素属于 A 当且仅当属于 B，因此集合相等。

Use / 怎么用

证明集合恒等式时最稳的方法是“取任意元素 x”：先证明 $x \in A \Rightarrow x \in B$，再证明 $x \in B \Rightarrow x \in A$。

模板：证明集合相等 / Proving Set Equality

写法：取任意 x。证明 $x \in A \Rightarrow x \in B$，再证明 $x \in B \Rightarrow x \in A$。最后说因此 $A = B$。

2.2 集合运算恒等式 / Set Identities

Theorem / 定理

集合运算与逻辑运算一一对应：$\cap$ 对应 $\land$，$\cup$ 对应 $\lor$，补集对应 $\neg$。

$$\begin{aligned}
A \cup \varnothing &= A, & A \cap U &= A,\\
A \cup U &= U, & A \cap \varnothing &= \varnothing,\\
A \cup A &= A, & A \cap A &= A,\\
(A^c)^c &= A, & &\\
A \cup B &= B \cup A, & A \cap B &= B \cap A,\\
A \cup (B \cap C) &= (A \cup B) \cap (A \cup C), & &\\
A \cap (B \cup C) &= (A \cap B) \cup (A \cap C), & &\\
(A \cap B)^c &= A^c \cup B^c, & &\\
(A \cup B)^c &= A^c \cap B^c, & &\\
A \cup (A \cap B) &= A, & A \cap (A \cup B) &= A,\\
A - B &= A \cap B^c, & &\\
A \oplus B &= (A - B) \cup (B - A). & &
\end{aligned} \tag{21}$$

Proof / 证明

用元素法。以德摩根律为例：

$$\begin{aligned}
x \in (A \cap B)^c &\iff x \notin A \cap B\\
&\iff \neg(x \in A \land x \in B)\\
&\iff x \notin A \lor x \notin B\\
&\iff x \in A^c \cup B^c.
\end{aligned} \tag{22}$$

Use / 怎么用

集合恒等式题常用两种方法：元素法更严谨，代数法更快。考试中如果题目要求证明，建议写元素法。

课堂展开：集合证明：元素法完整写法 / Set Proofs by Element Chasing

Example / 例题

证明：

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C). \tag{23}$$

Proof / 证明

先证左边包含右边。取任意 $x \in A \cap (B \cup C)$。由交集定义，$x \in A$ 且 $x \in B \cup C$。由并集定义，$x \in B$ 或 $x \in C$。

若 $x \in B$，则 $x \in A \cap B$，所以 $x \in (A \cap B) \cup (A \cap C)$。若 $x \in C$，则 $x \in A \cap C$，同样 $x \in (A \cap B) \cup (A \cap C)$。因此：

$$A \cap (B \cup C) \subseteq (A \cap B) \cup (A \cap C). \tag{24}$$

再证右边包含左边。取任意 $x \in (A \cap B) \cup (A \cap C)$。则 $x \in A \cap B$ 或 $x \in A \cap C$。第一种情况给出 $x \in A$ 且 $x \in B$；第二种情况给出 $x \in A$ 且 $x \in C$。无论哪种情况，都有 $x \in A$ 且 $x \in B \cup C$，所以 $x \in A \cap (B \cup C)$。因此：

$$(A \cap B) \cup (A \cap C) \subseteq A \cap (B \cup C). \tag{25}$$

两边互相包含，故相等。

How to use / 怎么用

集合恒等式证明不要只写“显然”。标准写法永远是：取 x，翻译定义，分类讨论，最后互相包含。

深讲：集合代数和逻辑代数的对应 / Sets as Logic

Correspondence / 对应关系

集合恒等式本质上是逻辑恒等式。对任意元素 x：

$$x \in A \cap B \iff (x \in A) \wedge (x \in B), \tag{26}$$

$$x \in A \cup B \iff (x \in A) \vee (x \in B), \tag{27}$$

$$x \in A^c \iff \neg(x \in A). \tag{28}$$

因此任何逻辑等价都能翻译成集合等价。

Example / 例题：证明第二条德摩根律

$$(A \cup B)^c = A^c \cap B^c. \tag{29}$$

取任意 x：

$$\begin{aligned} x \in (A \cup B)^c &\iff x \notin A \cup B \\ &\iff \neg(x \in A \vee x \in B) \\ &\iff (x \notin A) \wedge (x \notin B) \\ &\iff x \in A^c \cap B^c. \end{aligned} \tag{30}$$

所以两个集合有完全相同的元素，故相等。

How to use / 怎么用

如果题目要求证明集合恒等式，写元素链式等价最清楚。每一行都对应一个定义或逻辑等价式。

深讲：幂集、笛卡尔积和二进制选择 / Power Sets and Binary Choices

Theorem / 幂集大小

若 $|A| = n$，则：

$$|\mathcal{P}(A)| = 2^n. \tag{32}$$

Detailed proof / 详细证明

设 $A=\{a_1,a_2,\dots,a_n\}$。一个子集 $S\subseteq A$ 的形成过程是：对每个元素 a_i，选择“放入 S”或“不放入 S”。每个元素 2 种选择，且选择彼此独立，所以总数是：

$$2\cdot 2\cdots 2=2^n. \tag{33}$$

也可把每个子集对应到一个长度为 n 的 0-1 串：第 i 位为 1 表示 $a_i\in S$，为 0 表示 $a_i\notin S$。这是一个双射，所以子集数等于 0-1 串数，即 2^n。

How to use / 怎么用

看到“所有子集”“选择任意若干元素”“每个元素可选可不选”，通常就是 2^n。如果要求非空子集，就是 2^n-1。

2.3 基数与容斥 / Cardinality and Inclusion-Exclusion

Basic formulas / 基本公式

$$\begin{aligned}
&|\mathcal{P}(A)|=2^{|A|},\\
&|A\times B|=|A|\,|B|,\\
&|A\cup B|=|A|+|B|-|A\cap B|,\\
&|A-B|=|A|-|A\cap B|.
\end{aligned} \tag{34}$$

Theorem / 三集合容斥

$$|A\cup B\cup C|=|A|+|B|+|C|-|A\cap B|-|A\cap C|-|B\cap C|+|A\cap B\cap C|. \tag{35}$$

Theorem / 一般容斥原理

$$\left|\bigcup_{i=1}^{n}A_i\right|=\sum_i|A_i|-\sum_{i<j}|A_i\cap A_j|+\sum_{i<j<k}|A_i\cap A_j\cap A_k|-\cdots+(-1)^{n+1}|A_1\cap\cdots\cap A_n|. \tag{36}$$

Proof / 证明

考虑某个元素 x 恰好属于 r 个集合。右边对它的计数贡献为：

$$\binom{r}{1}-\binom{r}{2}+\binom{r}{3}-\cdots+(-1)^{r+1}\binom{r}{r}=1. \tag{37}$$

这是由 $0=(1-1)^r=\sum_{k=0}^{r}(-1)^k\binom{r}{k}$ 得到。若 x 不属于任何集合，贡献为 0。所以每个并集中的元素恰好被计数一次。

Use / 怎么用

看到“至少满足一个条件”“没有任何坏性质”“不能被若干数整除”等题型，先设坏集合或好集合，再用容斥。

Pitfall / 易错点

三集合容斥最后一项是加 $|A\cap B\cap C|$，因为三重交集在减去两两交集时被减多了。

课堂展开：容斥：从“至少一个”到“没有坏事件” / Inclusion-Exclusion in Detail

Example / 例题

求 1 到 100 中能被 2、3 或 5 整除的整数个数。

Solution / 解法

设：

$$A_2=\{n:2\mid n\},\quad A_3=\{n:3\mid n\},\quad A_5=\{n:5\mid n\}. \tag{38}$$

先算单个集合：

$$|A_2| = \left\lfloor \frac{100}{2} \right\rfloor = 50, \quad |A_3| = \left\lfloor \frac{100}{3} \right\rfloor = 33, \quad |A_5| = \left\lfloor \frac{100}{5} \right\rfloor = 20. \tag{39}$$

两两交集表示被最小公倍数整除：

$$|A_2 \cap A_3| = \left\lfloor \frac{100}{6} \right\rfloor = 16, \quad |A_2 \cap A_5| = \left\lfloor \frac{100}{10} \right\rfloor = 10, \quad |A_3 \cap A_5| = \left\lfloor \frac{100}{15} \right\rfloor = 6. \tag{40}$$

三重交集：

$$|A_2 \cap A_3 \cap A_5| = \left\lfloor \frac{100}{30} \right\rfloor = 3. \tag{41}$$

所以：

$$|A_2 \cup A_3 \cup A_5| = 50 + 33 + 20 - 16 - 10 - 6 + 3 = 74. \tag{42}$$

How to use / 怎么用

“能被若干数之一整除”是“至少满足一个条件”，直接用容斥。“不能被任何一个整除”则用补集：总数减去并集大小。

Common mistake / 常见错误

两两交集不是把两个计数相乘，而是同时满足两个整除条件；要用最小公倍数。

2.4 函数 / Functions

Definition / 定义

函数 $f : A \to B$ 给每个 $a \in A$ 指定唯一的 $f(a) \in B$。A 是定义域，B 是陪域，$f(A)$ 是值域。

Key notions / 核心概念

$$\begin{aligned} \text{injective} &\iff f(a_1) = f(a_2) \Rightarrow a_1 = a_2, \\ \text{surjective} &\iff \forall b \in B \, \exists a \in A \, f(a) = b, \\ \text{bijective} &\iff \text{injective and surjective.} \end{aligned} \tag{43}$$

Theorem / 有限集合上的鸽巢版本

若 A 和 B 有限且 $|A| = |B|$，则 $f : A \to B$ 单射当且仅当满射。

Proof / 证明

若 f 单射，则 A 中不同元素有不同像，所以 $|f(A)| = |A| = |B|$，因此 $f(A) = B$，满射。反向同理：若满射但不是单射，则两个不同元素映到同一点，值域大小至多 $|A| - 1$，无法覆盖 B。

Use / 怎么用

有限集合上证明双射时，只需证明单射或满射之一。计数题中常用双射把难数对象转成好数对象。

模板：证明函数单射 / Proving Injectivity

写法：设 $f(a) = f(b)$。通过代数化简推出 $a = b$。

模板：证明函数满射 / Proving Surjectivity

写法：取任意 y 属于陪域。解 $f(x) = y$ 得候选 x，验证 x 在定义域内。

课堂展开：函数：单射、满射、双射的完整证明 / Functions in Detail

Example 1 / 证明单射

令 $f : \mathbb{Z} \to \mathbb{Z}$，$f(n) = 3n + 2$。证明 f 是单射。

Proof / 证明

设 $f(a)=f(b)$。则：

$$3a+2=3b+2. \tag{44}$$

两边减 2 得 $3a=3b$，再除以 3 得 $a=b$。所以 f 是单射。

Example 2 / 判断满射

同一个函数 $f:\mathbb{Z}\to\mathbb{Z}$，$f(n)=3n+2$ 不是满射。

Proof / 证明

若 y 在值域中，则存在整数 n 使 $y=3n+2$，所以 $y\equiv 2 \pmod 3$。但 $0\in\mathbb{Z}$ 且 $0\not\equiv 2 \pmod 3$，没有整数 n 使 $3n+2=0$。因此不是满射。

How to use / 怎么用

证明满射时要从任意 y 出发解 $y=f(x)$；证明不是满射时只需要找一个陪域元素没有原像。

2.5 反函数与复合 / Inverses and Composition

Theorem / 反函数存在条件

函数 $f:A\to B$ 有反函数 $f^{-1}:B\to A$ 当且仅当 f 是双射。

Proof / 证明

若反函数存在，则 $f^{-1}(f(a))=a$ 保证单射，$f(f^{-1}(b))=b$ 保证满射。反过来，若 f 双射，每个 $b\in B$ 有唯一的 a 满足 $f(a)=b$，可定义 $f^{-1}(b)=a$。

Use / 怎么用

要构造反函数，先解方程 $y=f(x)$ 得 x 关于 y 的表达式，再检查定义域和陪域。

2.6 地板与天花板函数 / Floor and Ceiling

Formulas / 公式

$$\begin{aligned}
&\lfloor x\rfloor \le x < \lfloor x\rfloor + 1,\\
&\lceil x\rceil - 1 < x \le \lceil x\rceil,\\
&\lceil x\rceil = -\lfloor -x\rfloor,\\
&\lfloor x+n\rfloor = \lfloor x\rfloor + n \quad (n\in\mathbb{Z}),\\
&\left\lfloor \frac{n}{d}\right\rfloor = \text{number of positive multiples of } d \text{ not exceeding } n.
\end{aligned} \tag{45}$$

Proof / 证明

地板函数定义为不超过 x 的最大整数，因此第一组不等式直接来自最大性。天花板函数可由地板函数对称得到。

Use / 怎么用

整除计数、算法迭代次数、二分查找层数常用 $\lfloor \log_2 n\rfloor$ 或 $\lceil \log_2 n\rceil$。

2.7 可数与不可数 / Countability

Definition / 定义

集合 A 可数，指 A 有限或存在双射 $A\leftrightarrow\mathbb{N}$。Countable means finite or countably infinite.

Theorem / 定理

$$\mathbb{Z},\ \mathbb{Q},\ \mathbb{N}\times\mathbb{N}\quad \text{are countable.} \tag{46}$$

Proof / 证明

$\mathbb{Z}$ 可按 $0,1,-1,2,-2,...$ 枚举。$\mathbb{N}\times\mathbb{N}$ 可按对角线 $i+j=k$ 枚举。$\mathbb{Q}$ 可写成 a/b，其中 $a\in\mathbb{Z}$、$b\in\mathbb{N}^+$，它嵌入到可数集合 $\mathbb{Z}\times\mathbb{N}^+$ 中，因此可数。

Theorem / Cantor 对角线定理

$\mathbb{R}$ 不可数，甚至区间 $(0,1)$ 不可数。

Proof / 证明

假设 $(0,1)$ 中实数能列成 $r_1,r_2,...$。写十进制展开，构造新数 x 的第 i 位不同于 r_i 的第 i 位，并避免使用 9 与 0 的歧义。则 x 与每个 r_i 至少在第 i 位不同，所以不在列表中，矛盾。

Theorem / 代数数可数，超越数不可数

所有整系数多项式可数，每个非零多项式根有限，所以代数数可数。由于实数不可数，实数中除去可数个代数数后仍不可数，因此超越数不可数。

Use / 怎么用

证明可数：给枚举、给到 $\mathbb{N}$ 的编码，或证明是可数并。证明不可数：常用对角线法或构造从已知不可数集合的单射。

补充：Cantor 定理与 Schröder-Bernstein 定理 / Cantor and Schröder-Bernstein

Cantor theorem / Cantor 定理

对任意集合 A，不存在从 A 到 $\mathcal{P}(A)$ 的满射。因此：

$$|A|<|\mathcal{P}(A)|. \tag{47}$$

Proof / 证明

反设存在满射 $f:A\to\mathcal{P}(A)$。构造集合：

$$B=\{a\in A:a\notin f(a)\}. \tag{48}$$

因为 f 满射，存在 $b\in A$ 使 $f(b)=B$。现在问 $b\in B$ 是否成立。若 $b\in B$，由 B 的定义得 $b\notin f(b)=B$，矛盾。若 $b\notin B$，由 B 的定义得 $b\in f(b)=B$，也矛盾。因此满射不存在。

Use / 怎么用

这是证明不可数性的基本工具。实数不可数可以通过 $\mathcal{P}(\mathbb{N})$ 与二进制序列联系起来理解。

Schröder-Bernstein theorem / Schröder-Bernstein 定理

若存在单射 $f:A\to B$ 和单射 $g:B\to A$，则存在双射 $h:A\to B$，所以：

$$|A|=|B|. \tag{49}$$

Proof idea / 证明思路

完整证明要把 A 中元素按是否来自 g 的链分层，然后在某些链上用 f，在另一些链上用 g^{-1} 拼出双射。课堂中通常更重视会用它：要证明两个无限集合等势，可以分别构造两个方向的单射。

深讲：可数性的三种证明方式 / Three Ways to Prove Countability

Method 1: Explicit list / 直接列举

若能把元素排成序列 $a_0,a_1,a_2,...$，并保证每个元素最终出现，就证明了可数。

Method 2: Encoding / 编码到自然数

把对象编码成有限字符串或自然数。例如所有有限二进制串可按长度排序：长度 0、长度 1、长度 2，每层有限，所以总共可数。

Method 3: Countable union / 可数并

如果每个 A_i 可数，则：

$$\bigcup_{i=0}^{\infty} A_i \tag{50}$$

也是可数。证明方法是对二维表按对角线枚举。

Detailed proof that $\mathbb{Q}$ is countable / 有理数可数详细证明

每个正有理数可写成 p/q，其中 $p, q \in \mathbb{N}^+$。把所有 pair (p, q) 放进无限表格。按 $p + q = 2, 3, 4, \ldots$ 的对角线枚举，每条对角线有限。遇到不是最简分数的可以跳过。这样每个正有理数都会出现。再穿插 0 和负有理数，得到全部 $\mathbb{Q}$ 可数。

为什么实数不同 / Why $\mathbb{R}$ is different

对角线法不是说“当前列表漏了一个数”而已，而是说任意列表都会漏一个数，所以不存在完整列表。这是不可数性的核心。

2.8 序列与求和 / Sequences and Summations

Common sums / 常用求和公式

$$\begin{aligned}
\sum_{i=1}^{n} i &= \frac{n(n+1)}{2}, \\
\sum_{i=1}^{n} i^2 &= \frac{n(n+1)(2n+1)}{6}, \\
\sum_{i=1}^{n} i^3 &= \left(\frac{n(n+1)}{2}\right)^2, \\
\sum_{i=0}^{n} r^i &= \frac{r^{n+1}-1}{r-1} \quad (r \neq 1), \\
\sum_{i=0}^{\infty} r^i &= \frac{1}{1-r} \quad (|r| < 1), \\
\sum_{i=1}^{n} \frac{1}{i} &= H_n.
\end{aligned} \tag{51}$$

Integral bounds / 积分估计

若 f 单调递减且非负，则：

$$\int_{1}^{n+1} f(x)\,dx \le \sum_{i=1}^{n} f(i) \le f(1) + \int_{1}^{n} f(x)\,dx. \tag{52}$$

特别地：

$$\ln(n+1) \le H_n \le 1 + \ln n. \tag{53}$$

Proof / 证明

常用矩形面积夹逼。对于递减函数，每个整数点的矩形可上下夹住曲线下面积。

Use / 怎么用

算法复杂度里看到调和和 $1 + 1/2 + \cdots + 1/n$，可直接估为 $\Theta(\log n)$。

深讲：求和公式的来源 / Where Summation Formulas Come From

Arithmetic sum / 等差求和

$$1+2+\cdots+n=\frac{n(n+1)}{2}. \tag{54}$$

把和正着写一遍、倒着写一遍：

$$\begin{aligned} S &= 1+2+\cdots+(n-1)+n,\\ S &= n+(n-1)+\cdots+2+1. \end{aligned} \tag{55}$$

两式相加，每一列都是 $n+1$，共有 n 列：

$$2S=n(n+1),\qquad S=\frac{n(n+1)}{2}. \tag{56}$$

Geometric sum / 等比求和

若 $r\neq 1$，令：

$$S=1+r+r^2+\cdots+r^n. \tag{57}$$

两边乘以 r：

$$rS=r+r^2+\cdots+r^{n+1}. \tag{58}$$

相减：

$$S-rS=1-r^{n+1}. \tag{59}$$

所以：

$$S=\frac{1-r^{n+1}}{1-r}=\frac{r^{n+1}-1}{r-1}. \tag{60}$$

Telescoping / 望远镜求和

若项可以写成 b_i-b_{i+1}，则中间项会抵消：

$$\sum_{i=1}^{n}(b_i-b_{i+1})=b_1-b_{n+1}. \tag{61}$$

How to use / 怎么用

看到相邻项相消，想 telescoping。看到固定比例，想 geometric。看到多项式求和，先想已知公式或用归纳证明。

2.9 矩阵与布尔矩阵 / Matrices and Boolean Matrices

Matrix multiplication / 矩阵乘法

若 A 是 $m\times n$ 矩阵，B 是 $n\times p$ 矩阵，则 $C=AB$ 的元素为：

$$c_{ij}=\sum_{k=1}^{n}a_{ik}b_{kj}. \tag{62}$$

Boolean product / 布尔积

关系矩阵中常用布尔积：

$$c_{ij}=\bigvee_{k=1}^{n}(a_{ik}\wedge b_{kj}). \tag{63}$$

Use / 怎么用

关系 R 的矩阵 M_R 满足：M_R^2 的布尔积表示长度为 2 的路径是否存在。传递闭包可由布尔矩阵幂或 Warshall 算法求。

3. 算法与增长阶 / Algorithms and Growth

3.1 算法的性质 / Algorithm Properties

Definition / 定义

算法是有限、明确、可执行的步骤序列。通常要求输入、输出、确定性、有限性、有效性。

Use / 怎么用

分析算法题时分两层：先证明正确性，再分析时间复杂度。Correctness and complexity are separate claims.

3.2 渐近符号 / Asymptotic Notation

Definitions / 定义

$$
\begin{aligned}
&f(n) = O(g(n)) \iff \exists C > 0, \exists n_0, \forall n \geq n_0,\ 0 \leq f(n) \leq Cg(n),\\
&f(n) = \Omega(g(n)) \iff \exists C > 0, \exists n_0, \forall n \geq n_0,\ 0 \leq Cg(n) \leq f(n),\\
&f(n) = \Theta(g(n)) \iff f(n) = O(g(n)) \land f(n) = \Omega(g(n)),\\
&f(n) = o(g(n)) \iff \lim_{n\to\infty} \frac{f(n)}{g(n)} = 0,\\
&f(n) = \omega(g(n)) \iff \lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty.
\end{aligned}
\tag{64}
$$

Theorem / 极限判别

若 $g(n) > 0$ 且

$$
L = \lim_{n\to\infty} \frac{f(n)}{g(n)},
\tag{65}
$$

则：

- $0 < L < \infty$ implies $f(n) = \Theta(g(n))$.
- $L = 0$ implies $f(n) = o(g(n))$.
- $L = \infty$ implies $f(n) = \omega(g(n))$.

Proof / 证明

由极限定义。若 L 是正有限数，则当 n 足够大时，$f(n)/g(n)$ 被两个正数夹住。

Use / 怎么用

比较复杂函数时优先用极限。证明 Big-O 时必须给出具体常数 C 与阈值 n_0。

课堂展开：渐近符号：常数和阈值怎么写 / Big-O Proofs with Constants

Example / 例题

证明：

$$
3n^2 + 5n + 7 = O(n^2).
\tag{66}
$$

Proof / 证明

当 $n \geq 1$ 时，$5n \leq 5n^2$ 且 $7 \leq 7n^2$。所以：

$$
3n^2 + 5n + 7 \leq 3n^2 + 5n^2 + 7n^2 = 15n^2.
\tag{67}
$$

取 $C = 15$，$n_0 = 1$，由 Big-O 定义可得 $3n^2 + 5n + 7 = O(n^2)$。

Example / 反例

证明 $n^2 \neq O(n)$。

若 $n^2 = O(n)$，则存在 C, n_0 使 $n^2 \leq Cn$ 对所有 $n \geq n_0$ 成立。由于 $n > 0$，可除以 n 得 $n \leq C$ 对所有足够大的 n 成立，这不可能。取 $n > C$ 即矛盾。

How to use / 怎么用

- 证明 O：找上界常数。
- 证明 Ω：找下界常数。
- 证明 Θ：上下界都要写。
- 证明不是 O：反证，并让 n 超过任何固定常数。

3.3 常见增长阶 / Common Growth Hierarchy

Hierarchy / 层级

$$1 \prec \log\log n \prec \log n \prec n^c \prec a^n \prec n! \prec n^n, \tag{68}$$

其中 $c > 0$，$a > 1$。

更细版本：

$$\log^k n \prec n^\epsilon \prec n^c \prec n^c \log^k n \prec n^{c+\epsilon} \prec a^n. \tag{69}$$

Use / 怎么用

排序增长速度、判断循环嵌套、递归树求和。多项式总是慢于指数，任意固定幂的对数都慢于任意正幂的多项式。

3.4 常见算法复杂度 / Common Algorithm Complexities

Examples / 例子

- 线性搜索 / linear search: worst-case $O(n)$.
- 二分搜索 / binary search: $O(\log n)$ comparisons.
- 插入排序 / insertion sort: worst-case $O(n^2)$.
- 归并排序 / merge sort: $O(n \log n)$.
- 穷举子集 / enumerate subsets: $O(2^n)$.
- 穷举排列 / enumerate permutations: $O(n!)$.

Proof idea / 证明思路

二分搜索每次把候选规模减半，最多需要 k 次使 $n/2^k \leq 1$，即 $k \geq \log_2 n$。

Use / 怎么用

看到“每次减半”就是对数；看到“双重循环都到 n”通常是平方；看到“枚举所有子集”就是 2^n。

深讲：算法复杂度：循环如何翻译成求和 / Loops as Sums

Single loop / 单循环

如果循环执行 n 次，每次 $O(1)$，总复杂度是 $O(n)$。

Nested loop with triangular range / 三角形双循环

例如：外层 $i=1$ 到 n，内层 $j=1$ 到 i。总次数：

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2} = \Theta(n^2). \tag{70}$$

Logarithmic loop / 对数循环

若变量每次乘 2：$1, 2, 4, 8, \ldots$，直到超过 n，执行次数为最小的 k 使 $2^k > n$，所以是 $\Theta(\log n)$。

Harmonic loop / 调和型

如果第 i 层代价约为 n/i，总和是：

$$\sum_{i=1}^{n} \frac{n}{i} = nH_n = \Theta(n \log n). \tag{71}$$

How to use / 怎么用

算法题不要凭视觉判断几重循环。先写出执行次数求和，再化简渐近阶。

4. 数论与密码 / Number Theory and Cryptography

4.1 整除 / Divisibility

Definition / 定义

若存在整数 k 使 $b = ak$，则 a 整除 b，记作 $a \mid b$。

Theorem / 基本性质

$$\begin{aligned} a \mid b \wedge a \mid c &\Longrightarrow a \mid (b+c), \\ a \mid b &\Longrightarrow a \mid bc, \\ a \mid b \wedge b \mid c &\Longrightarrow a \mid c, \\ a \mid b \wedge a \mid c &\Longrightarrow a \mid (mb+nc) \quad (m, n \in \mathbb{Z}). \end{aligned} \tag{72}$$

Proof / 证明

若 $b = ak$、$c = a\ell$，则 $mb + nc = a(mk + n\ell)$，括号内是整数，所以 $a \mid (mb+nc)$。

Use / 怎么用

整除证明几乎都回到定义：把目标写成“某个整数乘以除数”。

模板：证明整除 / Proving Divisibility

写法：要证 $d \mid n$，就把 n 写成 dk，并说明 $k \in \mathbb{Z}$。

4.2 除法算法 / Division Algorithm

Theorem / 定理

对任意整数 a 和正整数 d，存在唯一整数 q, r 使：

$$a = dq + r, \qquad 0 \le r < d. \tag{73}$$

Proof / 证明

存在性可用良序原理：考虑集合 $S = \{a - dq \mid q \in \mathbb{Z},\ a - dq \ge 0\}$，取最小元素 r。若 $r \ge d$，则 $r - d$ 是更小非负元素，矛盾。唯一性：若 $a = dq + r = dq' + r'$ 且 $0 \le r, r' < d$，则 $d(q - q') = r' - r$。右侧绝对值小于 d，只能为 0，所以 $r = r'$、$q = q'$。

Use / 怎么用

它是模运算、欧几里得算法、进制表示的基础。题中出现“余数”或“mod”时默认使用这个定理。

深讲：除法算法和模运算的底层逻辑 / Division Algorithm Deep Dive

Why quotient and remainder exist / 为什么商和余数存在

对 a 除以正整数 d，考虑所有非负数 $a-dq$。这些数中有最小值 r。若 $r \geq d$，则 $r-d$ 仍非负且更小，矛盾。因此 $0 \leq r < d$。

Why they are unique / 为什么唯一

若：

$$a = dq + r = dq' + r', \qquad 0 \leq r, r' < d, \tag{74}$$

则：

$$d(q - q') = r' - r. \tag{75}$$

左边是 d 的倍数，而右边严格在 $-(d-1)$ 到 $d-1$ 之间。唯一可能的 d 的倍数是 0，所以 $r = r'$，进而 $q = q'$。

Connection to modular arithmetic / 和同余的关系

$a \bmod d$ 就是这个唯一余数。同余 $a \equiv b \pmod{d}$ 表示 a 和 b 除以 d 的余数相同。

4.3 同余 / Modular Arithmetic

Definition / 定义

$$a \equiv b \pmod{m} \iff m \mid (a-b). \tag{76}$$

Theorem / 同余运算规则

若 $a \equiv b \pmod{m}$ 且 $c \equiv d \pmod{m}$，则：

$$\begin{aligned} a + c &\equiv b + d \pmod{m}, \\ a - c &\equiv b - d \pmod{m}, \\ ac &\equiv bd \pmod{m}, \\ a^k &\equiv b^k \pmod{m} \quad (k \in \mathbb{N}). \end{aligned} \tag{77}$$

Proof / 证明

由定义，$m \mid (a-b)$ 且 $m \mid (c-d)$。加减法由整除线性组合性质得到。乘法：

$$ac - bd = ac - bc + bc - bd = c(a-b) + b(c-d), \tag{78}$$

右边被 m 整除。

Use / 怎么用

计算大数余数时，先把每个因子或幂降到小余数，再运算。例如 $7^{100} \bmod 10$ 可以利用周期。

模板：证明同余 / Proving Congruence

写法：要证 $a \equiv b \pmod{m}$，就证 $m \mid (a-b)$，或把 a, b 都化成相同余数。

4.4 最大公约数与欧几里得算法 / GCD and Euclidean Algorithm

Definitions / 定义

$\gcd(a,b)$ 是同时整除 a 和 b 的最大正整数。$\operatorname{lcm}(a,b)$ 是最小正公倍数。

Theorem / 欧几里得递推

若 $a = bq + r$，则：

$$\gcd(a, b) = \gcd(b, r). \tag{79}$$

Proof / 证明

任意公因子 d 同时整除 a 和 b 当且仅当它整除 b 和 $a - bq = r$。所以两边公因子集合相同，最大值相同。

Formula / 公式

$$\gcd(a, b)\,\mathrm{lcm}(a, b) = |ab|. \tag{80}$$

Proof / 证明

用素因数分解。若 $a = \prod p_i^{\alpha_i}$、$b = \prod p_i^{\beta_i}$，则：

$$\gcd(a, b) = \prod p_i^{\min(\alpha_i, \beta_i)}, \qquad \mathrm{lcm}(a, b) = \prod p_i^{\max(\alpha_i, \beta_i)}. \tag{81}$$

指数相加为 $\alpha_i + \beta_i$。

Use / 怎么用

求 gcd 用欧几里得算法；求 lcm 可先求 gcd 再用公式，避免直接列倍数。

4.5 Bezout 定理与扩展欧几里得 / Bezout and Extended Euclidean Algorithm

Theorem / Bezout identity

对整数 a, b，存在整数 s, t 使：

$$\gcd(a, b) = sa + tb. \tag{82}$$

Proof / 证明

用欧几里得算法反代。最后一个非零余数是 gcd，而每个余数都是前两个数的整数线性组合，所以 gcd 也是 a, b 的整数线性组合。

Corollary / 推论

$$\gcd(a, b) = 1 \quad \Longleftrightarrow \quad \exists s, t \in \mathbb{Z},\ sa + tb = 1. \tag{83}$$

Use / 怎么用

求模逆元：若 $\gcd(a, m) = 1$ 且 $sa + tm = 1$，则 $sa \equiv 1 \pmod{m}$，所以 s 是 a 模 m 的逆元。

课堂展开：数论：欧几里得算法与 Bezout 反代 / Euclidean Algorithm in Detail

Example / 例题

求 $\gcd(252, 198)$，并写成 252 和 198 的整数线性组合。

Solution / 解法

先做欧几里得算法：

$$\begin{aligned} 252 &= 1 \cdot 198 + 54, \\ 198 &= 3 \cdot 54 + 36, \\ 54 &= 1 \cdot 36 + 18, \\ 36 &= 2 \cdot 18 + 0. \end{aligned} \tag{84}$$

最后一个非零余数是 18，所以：

$$\gcd(252, 198) = 18. \tag{85}$$

反代求 Bezout 表达：

$$\begin{aligned} 18 &= 54 - 1 \cdot 36, \\ 36 &= 198 - 3 \cdot 54. \end{aligned} \tag{86}$$

代入：

$$\begin{aligned} 18 &= 54 - (198 - 3 \cdot 54) \\ &= 4 \cdot 54 - 198. \end{aligned} \tag{87}$$

又 $54 = 252 - 198$，所以：

$$18 = 4(252 - 198) - 198 = 4 \cdot 252 - 5 \cdot 198. \tag{88}$$

How to use / 怎么用

求模逆、解线性同余、证明互素时都需要 Bezout。反代的原则是从最后一个非零余数一路往上替换。

4.6 素数与唯一分解 / Primes and Fundamental Theorem of Arithmetic

Definition / 定义

大于 1 的整数 p 若正因子只有 1 和 p，则 p 是素数。

Euclid's Lemma / 欧几里得引理

若 p 是素数且 $p \mid ab$，则 $p \mid a$ 或 $p \mid b$。

Proof / 证明

若 $p \nmid a$，则 $\gcd(p, a) = 1$。由 Bezout，有 $sp + ta = 1$。两边乘以 b 得 $spb + tab = b$。由于 $p \mid spb$ 且 $p \mid tab$，所以 $p \mid b$。

Theorem / 算术基本定理

每个大于 1 的整数都能唯一写成素数乘积，忽略因子顺序。

$$n = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_k^{\alpha_k}. \tag{89}$$

Proof / 证明

存在性用强归纳：若 n 是素数，已完成；若合数，写成 ab 且 $1 < a, b < n$，由归纳假设分解 a, b。唯一性用欧几里得引理：若两种素数分解相等，则左侧某个素数整除右侧乘积，必整除某个右侧素数，因此两者相同，约去后归纳。

Use / 怎么用

gcd、lcm、整除判断、因子个数都可由素因数分解解决。若 $n = \prod p_i^{\alpha_i}$，正因子个数为：

$$\tau(n) = \prod_{i=1}^{k} (\alpha_i + 1). \tag{90}$$

补充：素数判定与素数无穷性 / Extra Prime Theorems

Infinitely many primes / 素数有无穷多个

素数有无穷多个。

Proof / 证明

反设只有有限多个素数 $p_1, p_2, \dots, p_k$。令：

$$N = p_1 p_2 \cdots p_k + 1. \tag{91}$$

对每个 p_i，都有 $N \equiv 1 \pmod{p_i}$，所以没有任何 p_i 整除 N。但 $N > 1$ 必有素因子，这个素因子不在列表中，矛盾。

Trial division theorem / 试除定理

若 n 是合数，则 n 有一个不超过 $\sqrt{n}$ 的素因子。

Proof / 证明

若 $n = ab$ 且 $1 < a \le b < n$，则 $a^2 \le ab = n$，所以 $a \le \sqrt{n}$。再取 a 的一个素因子 p，有 $p \le a \le \sqrt{n}$。

Use / 怎么用

判断 n 是否素数，只需试除所有不超过 $\sqrt{n}$ 的素数。

4.7 模逆元与线性同余 / Modular Inverses and Linear Congruences

Theorem / 模逆元存在条件

a 在模 m 下有乘法逆元，当且仅当 $\gcd(a, m) = 1$。

Proof / 证明

若 $ax \equiv 1 \pmod{m}$，则存在 y 使 $ax + my = 1$，所以 $\gcd(a, m) = 1$。反过来由 Bezout，若 $sa + tm = 1$，则 $sa \equiv 1 \pmod{m}$。

Theorem / 线性同余

方程 $ax \equiv b \pmod{m}$ 有解当且仅当 $d = \gcd(a, m)$ 整除 b。若有解，则模 m 下恰有 d 个不同解。

Proof / 证明

$ax \equiv b \pmod{m}$ 等价于 $ax + my = b$。线性组合 $ax + my$ 的所有可能值正是 d 的倍数，所以可解当且仅当 $d \mid b$。除以 d 后得到与 m/d 互素的系数，因此有唯一解模 m/d，提升到模 m 给出 d 个解。

Use / 怎么用

解法步骤：先算 $d = \gcd(a, m)$；若 $d \nmid b$ 无解；若可解，除以 d，求逆元，最后写出 d 个模 m 的解。

课堂展开：模逆元与线性同余：完整流程 / Modular Inverses and Linear Congruences

Example 1 / 求模逆

求 7 在模 26 下的逆元。

Solution / 解法

用欧几里得算法：

$$26 = 3 \cdot 7 + 5, \quad 7 = 1 \cdot 5 + 2, \quad 5 = 2 \cdot 2 + 1. \tag{92}$$

反代：

$$\begin{aligned} 1 &= 5 - 2 \cdot 2 \\ &= 5 - 2(7 - 5) \\ &= 3 \cdot 5 - 2 \cdot 7 \\ &= 3(26 - 3 \cdot 7) - 2 \cdot 7 \\ &= 3 \cdot 26 - 11 \cdot 7. \end{aligned} \tag{93}$$

所以：

$$-11 \cdot 7 \equiv 1 \pmod{26}. \tag{94}$$

因此 $7^{-1} \equiv -11 \equiv 15 \pmod{26}$。

Example 2 / 解线性同余

解：

$$14x \equiv 30 \pmod{100}. \tag{95}$$

先算 $d = \gcd(14, 100) = 2$。因为 $2 \mid 30$，有解。两边和模数同除以 2：

$$7x \equiv 15 \pmod{50}. \tag{96}$$

$7^{-1} \pmod{50}$ 可由 $50 = 7 \cdot 7 + 1$ 得到：

$$1 = 50 - 7 \cdot 7, \tag{97}$$

所以 $7^{-1} \equiv -7 \equiv 43 \pmod{50}$。于是：

$$x \equiv 15 \cdot 43 \equiv 645 \equiv 45 \pmod{50}. \tag{98}$$

回到模 100，有 $d = 2$ 个解：

$$x \equiv 45 \pmod{100}, \qquad x \equiv 95 \pmod{100}. \tag{99}$$

How to use / 怎么用

线性同余永远按三步：算 gcd；判断是否整除右边；除以 gcd 后求逆元。

4.8 中国剩余定理 / Chinese Remainder Theorem

Theorem / CRT

若 $m_1, \dots, m_k$ 两两互素，则系统

$$x \equiv a_i \pmod{m_i} \quad (i = 1, \dots, k) \tag{100}$$

在模 $M = m_1 m_2 \cdots m_k$ 下有唯一解。

Construction / 构造公式

令 $M_i = M/m_i$，取 y_i 满足 $M_i y_i \equiv 1 \pmod{m_i}$。则：

$$x \equiv \sum_{i=1}^{k} a_i M_i y_i \pmod{M}. \tag{101}$$

Proof / 证明

对固定 j，当 $i \neq j$ 时 M_i 被 m_j 整除，所以对应项模 m_j 为 0；当 $i = j$ 时 $M_j y_j \equiv 1 \pmod{m_j}$，因此总和模 m_j 等于 a_j。唯一性：若两个解同余于每个 m_i，它们差被每个 m_i 整除；两两互素，所以被 M 整除。

Use / 怎么用

CRT 用于把一个大模数问题拆成多个小模数问题，或把多个余数条件合并为一个同余类。

补充：广义中国剩余定理 / Generalized CRT

Theorem / 定理

同余组：

$$x \equiv a \pmod{m}, \qquad x \equiv b \pmod{n} \tag{102}$$

有解当且仅当：

$$a \equiv b \pmod{\gcd(m,n)}. \tag{103}$$

若有解，则解在模 $\operatorname{lcm}(m,n)$ 下唯一。

Proof / 证明

若 x 同时满足两式，则 $m \mid (x-a)$ 且 $n \mid (x-b)$。于是 $x \equiv a \pmod{d}$ 且 $x \equiv b \pmod{d}$，其中 $d = \gcd(m,n)$，所以 $a \equiv b \pmod{d}$。反向可把问题化成线性同余 $mt \equiv b-a \pmod{n}$，它有解当且仅当 $d \mid (b-a)$。

Use / 怎么用

普通 CRT 要求模数两两互素；广义 CRT 用来处理不互素模数。先检查余数在 gcd 意义下是否兼容。

课堂展开：中国剩余定理：构造法 / CRT by Construction

Example / 例题

解同余方程组：

$$\begin{aligned} x &\equiv 2 \pmod{3}, \\ x &\equiv 3 \pmod{5}, \\ x &\equiv 2 \pmod{7}. \end{aligned} \tag{104}$$

Solution / 解法

模数两两互素，$M = 3 \cdot 5 \cdot 7 = 105$。

$$M_1 = 35, \quad M_2 = 21, \quad M_3 = 15. \tag{105}$$

求逆元：

$$35y_1 \equiv 1 \pmod{3} \Rightarrow 2y_1 \equiv 1 \pmod{3} \Rightarrow y_1 = 2, \tag{106}$$

$$21y_2 \equiv 1 \pmod{5} \Rightarrow y_2 \equiv 1 \pmod{5}, \tag{107}$$

$$15y_3 \equiv 1 \pmod{7} \Rightarrow y_3 \equiv 1 \pmod{7}. \tag{108}$$

构造：

$$x \equiv 2 \cdot 35 \cdot 2 + 3 \cdot 21 \cdot 1 + 2 \cdot 15 \cdot 1 = 233 \equiv 23 \pmod{105}. \tag{109}$$

Check / 检查

$$23 \equiv 2 \pmod{3}, \quad 23 \equiv 3 \pmod{5}, \quad 23 \equiv 2 \pmod{7}. \tag{110}$$

How to use / 怎么用

CRT 计算一定要最后检查。构造公式看起来复杂，但每一项只负责在一个模数下留下目标余数，在其他模数下变成 0。

4.9 费马、欧拉与 Wilson / Fermat, Euler, and Wilson

Fermat's Little Theorem / 费马小定理

若 p 是素数且 $p \nmid a$，则：

$$a^{p-1} \equiv 1 \pmod{p}. \tag{111}$$

等价形式：对任意整数 a，$a^p \equiv a \pmod{p}$。

Proof / 证明

集合 $\{a, 2a, \dots, (p-1)a\}$ 模 p 后是 $\{1, 2, \dots, p-1\}$ 的重排。乘积同余：

$$a^{p-1}(p-1)! \equiv (p-1)! \pmod{p}. \tag{112}$$

因为 $(p-1)!$ 与 p 互素，可约去，得结论。

Euler phi / 欧拉函数

若 $n = \prod p_i^{\alpha_i}$，则：

$$\varphi(n) = n \prod_{p \mid n} \left(1 - \frac{1}{p}\right). \tag{113}$$

Euler's Theorem / 欧拉定理

若 $\gcd(a, n) = 1$，则：

$$a^{\varphi(n)} \equiv 1 \pmod{n}. \tag{114}$$

Proof / 证明

与费马小定理相同：把模 n 下与 n 互素的剩余类乘以 a，得到同一集合的重排。乘积后约去。

Wilson's Theorem / Wilson 定理

p 为素数当且仅当：

$$(p-1)! \equiv -1 \pmod{p}. \tag{115}$$

Proof / 证明

若 p 素数，模 p 下每个非零元素有逆元。除 1 和 $p-1$ 外，元素可与不同逆元配对，乘积为 1；剩下 $1 \cdot (p-1) \equiv -1$。反向若 n 合数，通常可找到非平凡因子使 $(n-1)!$ 被该因子整除，不可能同余 -1。

Use / 怎么用

费马/欧拉用于降幂取模；Wilson 多用于判断素数或处理阶乘模素数。

课堂展开：费马与欧拉：降幂取模 / Reducing Exponents

Example / 例题

计算：

$$7^{222} \pmod{13}. \tag{116}$$

Solution / 解法

因为 13 是素数且 $13 \nmid 7$，费马小定理给出：

$$7^{12} \equiv 1 \pmod{13}. \tag{117}$$

把指数除以 12：

$$222 = 18 \cdot 12 + 6. \tag{118}$$

所以：

$$7^{222} \equiv (7^{12})^{18} 7^6 \equiv 7^6 \pmod{13}. \tag{119}$$

继续算：

$$7^2 = 49 \equiv 10 \pmod{13}, \quad 7^4 \equiv 10^2 = 100 \equiv 9 \pmod{13}, \tag{120}$$

$$7^6 = 7^4 \cdot 7^2 \equiv 9 \cdot 10 = 90 \equiv 12 \pmod{13}. \tag{121}$$

答案是：

$$7^{222} \equiv 12 \pmod{13}. \tag{122}$$

How to use / 怎么用

先看模数是否素数。素数用费马；合数但底数互素用欧拉；底数不互素时要小心，常用 CRT 拆模数。

深讲：欧拉函数公式的详细证明 / Euler Phi Formula

Theorem / 定理

若：

$$n = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_k^{\alpha_k}, \tag{123}$$

则：

$$\varphi(n) = n \prod_{i=1}^{k} \left(1 - \frac{1}{p_i}\right). \tag{124}$$

Proof by inclusion-exclusion / 容斥证明

$\varphi(n)$ 计数的是 $1, 2, \ldots, n$ 中与 n 互素的整数。一个数不与 n 互素，当且仅当它被某个素因子 p_i 整除。

令 A_i 为 1 到 n 中被 p_i 整除的数。则：

$$|A_i| = \frac{n}{p_i}, \qquad |A_i \cap A_j| = \frac{n}{p_i p_j}, \tag{125}$$

更一般地：

$$|A_{i_1} \cap \cdots \cap A_{i_t}| = \frac{n}{p_{i_1} \cdots p_{i_t}}. \tag{126}$$

与 n 互素的数量是没有落入任何 A_i 的数量：

$$\begin{aligned} \varphi(n) &= n - \sum_i \frac{n}{p_i} + \sum_{i<j} \frac{n}{p_i p_j} - \cdots \\ &= n \prod_{i=1}^{k} \left(1 - \frac{1}{p_i}\right). \end{aligned} \tag{127}$$

How to use / 怎么用

算 $\varphi(n)$ 时先分解质因数，不要把所有小于 n 的数一个个检查。例如：

$$\varphi(60) = 60 \left(1 - \frac{1}{2}\right)\left(1 - \frac{1}{3}\right)\left(1 - \frac{1}{5}\right) = 16. \tag{128}$$

4.10 进制表示与快速幂 / Base Representation and Fast Exponentiation

Base expansion / 进制表示

对整数 $b > 1$，任意正整数 n 可唯一写成：

$$n = a_k b^k + a_{k-1} b^{k-1} + \cdots + a_1 b + a_0, \qquad 0 \le a_i < b, \qquad a_k \neq 0. \tag{129}$$

Proof / 证明

反复使用除法算法：$n = bq_0 + a_0$，再对 q_0 除以 b，直到商为 0。唯一性来自除法算法中余数唯一。

Fast exponentiation / 快速幂

若 n 的二进制表示为 $n=\sum b_i 2^i$，则：

$$a^n=\prod_{i:b_i=1} a^{2^i}. \tag{130}$$

Use / 怎么用

模幂计算中反复平方并取模，避免直接算巨大幂。复杂度为 $O(\log n)$ 次乘法。

4.11 RSA

Setup / 设置

选择大素数 p,q，令 $n=pq$，$\varphi(n)=(p-1)(q-1)$。选 e 满足 $\gcd(e,\varphi(n))=1$，求 d 使：

$$ed\equiv 1 \pmod{\varphi(n)}. \tag{131}$$

加密 $C\equiv M^e \pmod{n}$，解密 $M\equiv C^d \pmod{n}$。

Correctness proof / 正确性证明

因为 $ed=1+k\varphi(n)$。若 $\gcd(M,n)=1$，由欧拉定理：

$$(M^e)^d=M^{ed}=M^{1+k\varphi(n)}\equiv M \pmod{n}. \tag{132}$$

若 M 与 n 不互素，可分别模 p 和模 q 验证，再用 CRT 合并。

Use / 怎么用

RSA 题通常考：求 d、加密/解密小消息、解释为什么解密正确。核心是模逆元和欧拉定理。

深讲：费马、欧拉和 RSA 的证明链 / Fermat-Euler-RSA Chain

Permutation idea / 重排思想

费马小定理和欧拉定理的核心都是：如果 a 与模数互素，那么乘以 a 会把所有可逆剩余类重新排列。

对欧拉定理，设模 n 下所有与 n 互素的剩余类为：

$$r_1,r_2,\dots,r_{\varphi(n)}. \tag{133}$$

因为 $\gcd(a,n)=1$，所以：

$$ar_1,ar_2,\dots,ar_{\varphi(n)} \tag{134}$$

模 n 后仍是同一批剩余类的重排。于是乘积同余：

$$a^{\varphi(n)}r_1r_2\cdots r_{\varphi(n)}\equiv r_1r_2\cdots r_{\varphi(n)} \pmod{n}. \tag{135}$$

由于每个 r_i 都与 n 互素，乘积也与 n 互素，可以约去，得：

$$a^{\varphi(n)}\equiv 1 \pmod{n}. \tag{136}$$

RSA correctness / RSA 正确性更细版本

公钥指数 e 和私钥指数 d 满足：

$$ed=1+k\varphi(n). \tag{137}$$

解密时：

$$C^d \equiv (M^e)^d = M^{ed} = M^{1+k\varphi(n)}. \tag{138}$$

如果 $\gcd(M, n) = 1$，由欧拉定理：

$$M^{k\varphi(n)} \equiv 1. \tag{139}$$

所以：

$$M^{1+k\varphi(n)} \equiv M. \tag{140}$$

若 M 与 $n = pq$ 不互素，则分别模 p、模 q 讨论。若 $p \mid M$，则两边模 p 都为 0；若 $p \nmid M$，用费马小定理。对 q 同理。最后由 CRT 得到模 pq 下相同。

4.12 伪素数、原根、离散对数 / Pseudoprimes, Primitive Roots, Discrete Logs

Definitions / 定义

若合数 n 满足 $a^{n-1} \equiv 1 \pmod{n}$，则称 n 是以 a 为底的 Fermat 伪素数。

若 g 的幂生成模 p 的所有非零剩余类，则 g 是模 p 的原根：

$$\{g^1, g^2, \ldots, g^{p-1}\} \equiv \{1, 2, \ldots, p-1\} \pmod{p}. \tag{141}$$

离散对数问题是给定 g, h, p，求 x 使：

$$g^x \equiv h \pmod{p}. \tag{142}$$

Use / 怎么用

伪素数说明费马测试不是充分条件。原根和离散对数是 Diffie-Hellman 等密码协议的数学基础。

5. 归纳、递归与不变量 / Induction, Recursion, and Invariants

5.1 良序原理 / Well-Ordering Principle

Theorem / 定理

每个非空的非负整数集合都有最小元素。

Proof / 证明

在 Rosen 和 MIT 中，良序原理通常作为整数公理之一，或与数学归纳法等价。

Use / 怎么用

当要证明“不存在反例”时，假设有反例集合，取最小反例，再构造更小反例导致矛盾。

5.2 数学归纳法 / Mathematical Induction

Theorem / 定理

若命题 $P(n)$ 满足：

$$P(n_0) \quad \text{and} \quad \forall k \geq n_0 \, (P(k) \to P(k+1)), \tag{143}$$

则对所有 $n \geq n_0$，$P(n)$ 成立。

Proof / 证明

若存在反例，反例集合有最小元素 m。由于 $P(n_0)$ 成立，$m > n_0$。由最小性 $P(m-1)$ 成立，再由归纳步骤推出 $P(m)$，矛盾。

Use / 怎么用

归纳证明必须写：base case, induction hypothesis, induction step。常用于求和公式、整除、递推、图中边数、树的性质。

Example / 示例

证明：

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}. \tag{144}$$

基础步 $n = 1$ 成立。假设对 $n = k$ 成立，则：

$$\sum_{i=1}^{k+1} i = \frac{k(k+1)}{2} + (k+1) = \frac{(k+1)(k+2)}{2}. \tag{145}$$

课堂展开：归纳法：写完整 induction step / Writing Induction Clearly

Example / 例题

证明前 n 个奇数和为 n^2：

$$1 + 3 + 5 + \cdots + (2n-1) = n^2. \tag{146}$$

Proof / 证明

令 $P(n)$ 表示上述命题。

Base case: 当 $n = 1$ 时，左边为 1，右边为 $1^2 = 1$，成立。

Induction hypothesis: 假设 $P(k)$ 成立，即：

$$1 + 3 + \cdots + (2k-1) = k^2. \tag{147}$$

Induction step: 要证 $P(k+1)$：

$$1 + 3 + \cdots + (2k-1) + (2(k+1)-1). \tag{148}$$

由归纳假设，前 k 项和为 k^2，所以：

$$\begin{aligned} 1 + 3 + \cdots + (2k-1) + (2k+1) &= k^2 + 2k + 1 \\ &= (k+1)^2. \end{aligned} \tag{149}$$

所以 $P(k+1)$ 成立。由数学归纳法，对所有 $n \geq 1$ 成立。

How to use / 怎么用

归纳步骤中必须明确指出哪里用了归纳假设。不要只写“显然成立”。

5.3 强归纳法 / Strong Induction

Theorem / 定理

若 $P(n_0), \ldots, P(n_1)$ 成立，并且对每个 $k \geq n_1$，从 $P(n_0), \ldots, P(k)$ 可推出 $P(k+1)$，则所有 $n \geq n_0$ 都成立。

Proof / 证明

强归纳与普通归纳等价。令 $Q(n)$ 表示 $P(n_0), \ldots, P(n)$ 都成立，对 $Q(n)$ 做普通归纳。

Use / 怎么用

当 $P(k+1)$ 依赖多个更小情况而非只依赖 $P(k)$ 时用强归纳。典型：素因数分解、递归算法、硬币找零、图中路径分解。

课堂展开：强归纳：为什么可以假设所有更小情况 / Strong Induction

Example / 例题

证明每个大于 1 的整数都可以写成素数乘积。

Proof / 证明

对 $n \geq 2$ 做强归纳。

Base case: $n = 2$，它本身是素数，所以是素数乘积。

Induction hypothesis: 假设对所有整数 m，若 $2 \leq m \leq k$，则 m 可以写成素数乘积。

Induction step: 考虑 $k+1$。如果 $k+1$ 是素数，则已完成。如果 $k+1$ 是合数，则存在 a, b 满足：

$$k+1 = ab, \qquad 2 \leq a \leq k, \qquad 2 \leq b \leq k. \tag{150}$$

由强归纳假设，a 和 b 都可以写成素数乘积，因此 $ab = k+1$ 也可以写成素数乘积。

How to use / 怎么用

当当前对象会拆成两个或多个更小对象时，强归纳比普通归纳自然。素因数分解、递归算法、树结构都常用强归纳。

5.4 递归定义与递归算法 / Recursive Definitions and Algorithms

Examples / 例子

阶乘：

$$0! = 1, \qquad n! = n(n-1)! \quad (n \geq 1). \tag{151}$$

Fibonacci 数：

$$F_0 = 0, \quad F_1 = 1, \quad F_n = F_{n-1} + F_{n-2} \quad (n \geq 2). \tag{152}$$

Correctness proof / 正确性证明

递归算法正确性通常用归纳证明：基础输入直接正确；递归调用对更小输入正确；合并步骤保持正确。

Use / 怎么用

先确认递归会终止：每次调用都必须让某个自然数度量下降。Then prove correctness by induction on that measure.

5.5 结构归纳 / Structural Induction

Theorem / 定理

若递归定义的对象由基础对象和构造规则生成。证明性质 P 对所有对象成立，只需证明：

1. P 对基础对象成立。
2. 若 P 对构造规则的输入成立，则对构造出的新对象成立。

Proof / 证明

对象的生成深度是自然数。对生成深度做归纳即可。

Use / 怎么用

用于字符串、树、表达式、递归定义集合、布尔公式。

5.6 不变量 / Invariants

Definition / 定义

不变量是在状态变化过程中始终保持为真的性质。

Invariant proof / 不变量证明模板

$$\text{initial state satisfies } I \quad \wedge \quad (I \wedge \text{one step}) \Rightarrow I \quad \Longrightarrow \quad I \text{ holds forever.} \tag{153}$$

Proof / 证明

对步数做归纳。初始步为 base case；每次状态转移保持 I 是 induction step。

Use / 怎么用

MIT 特别强调状态机和不变量。常用于程序正确性、循环、游戏、图算法、并发协议。

6. 计数 / Counting

6.1 基本计数原则 / Basic Counting Principles

Sum rule / 加法原则

若选择可分成互不重叠的情况 $A_1, \dots, A_k$，则：

$$|A_1 \cup \cdots \cup A_k| = |A_1| + \cdots + |A_k|. \tag{154}$$

Product rule / 乘法原则

若一个过程分成 k 步，第 i 步有 n_i 种选择，且选择数不依赖于后续细节，则总数为：

$$n_1 n_2 \cdots n_k. \tag{155}$$

Division rule / 除法原则

若每个目标对象被某种计数方法恰好计数 d 次，则真实数量为：

$$\frac{\text{number of counted representations}}{d}. \tag{156}$$

Use / 怎么用

先判断情况是否互斥；互斥用加法，不互斥用容斥。多步独立选择用乘法；出现“顺序不重要”通常要除以重复排列数。

课堂展开：计数：先问“对象是什么” / Counting Starts from Objects

Counting decision tree / 计数决策树

1. 是否在数序列？若顺序重要，用乘法原则或排列。
2. 是否在数集合？若顺序不重要，用组合。
3. 是否允许重复？允许重复常用星棒法或 n^r。
4. 是否有禁止条件？少量禁止条件用补集或容斥。
5. 是否每个对象被数多次？用除法原则。

Example / 例题：MISSISSIPPI 排列数

MISSISSIPPI 有 11 个字母，其中：

$$M:1, \quad I:4, \quad S:4, \quad P:2. \tag{157}$$

不同排列数为：

$$\frac{11!}{1!4!4!2!}. \tag{158}$$

Why / 为什么

如果把所有相同字母先当成不同标签，有 11! 种排列。但 4 个 I 互换不改变字符串，4 个 S 互换不改变字符串，2 个 P 互换不改变字符串，所以要除以 4!4!2!。

6.2 排列组合 / Permutations and Combinations

Formulas / 公式

$$\begin{aligned} P(n,r) &= n(n-1)\cdots(n-r+1) = \frac{n!}{(n-r)!}, \\ \binom{n}{r} &= \frac{n!}{r!(n-r)!}, \\ \binom{n}{r} &= \binom{n}{n-r}. \end{aligned} \tag{159}$$

Proof / 证明

$P(n,r)$ 来自乘法原则：第一个位置 n 种，第二个 $n-1$ 种，依此类推。组合数由排列数除以所选 r 个元素内部的 $r!$ 个顺序。

Use / 怎么用

有顺序用排列 $P(n,r)$；无顺序用组合 $\binom{n}{r}$。先选人再排座位的题，通常组合和排列都会出现。

深讲：排列组合公式的证明细节 / Permutations and Combinations Deep Dive

Permutation / 排列公式

从 n 个不同对象中按顺序选 r 个。第 1 位有 n 种，第 2 位有 $n-1$ 种，直到第 r 位有 $n-r+1$ 种。所以：

$$P(n,r) = n(n-1)\cdots(n-r+1) = \frac{n!}{(n-r)!}. \tag{160}$$

Combination / 组合公式

无顺序选择 r 个对象。每个 r 元子集可以排列成 $r!$ 个有序列表。所以：

$$\binom{n}{r} = \frac{P(n,r)}{r!} = \frac{n!}{r!(n-r)!}. \tag{161}$$

Why $\binom{n}{r} = \binom{n}{n-r}$ / 对称性证明

选择 r 个元素等价于选择不被选中的 $n-r$ 个元素。映射 $S \mapsto A - S$ 是双射，因此数量相同。

How to use / 怎么用

如果选出来的位置有角色差异，用排列；如果只是一个集合，用组合。如果先选再排，通常是 $\binom{n}{r}r!$。

6.3 二项式系数恒等式 / Binomial Identities

Theorem / Pascal 恒等式

$$\binom{n}{r} = \binom{n-1}{r} + \binom{n-1}{r-1}. \tag{162}$$

Proof / 组合证明

从 n 个元素中选 r 个。固定一个特殊元素 x。若不选 x，有 $\binom{n-1}{r}$ 种；若选 x，还需从其余 $n-1$ 个中选 $r-1$ 个，有 $\binom{n-1}{r-1}$ 种。

Other identities / 其他恒等式

$$
\begin{aligned}
\sum_{r=0}^{n}\binom{n}{r} &= 2^n,\\
\sum_{r=0}^{n}(-1)^r\binom{n}{r} &= 0,\\
\sum_{r=0}^{n} r\binom{n}{r} &= n2^{n-1},\\
\sum_{k=r}^{n}\binom{k}{r} &= \binom{n+1}{r+1},\\
\sum_{k=0}^{r}\binom{m}{k}\binom{n}{r-k} &= \binom{m+n}{r}.
\end{aligned}
\tag{163}
$$

Proof / 证明思路

第一条：每个元素选或不选，共 2^n 个子集，也可按子集大小分类。第二条来自 $(1-1)^n$。第三条可数“从 n 人中选一个队长和若干队员”：左边先选队伍再选队长，右边先选队长，再决定其余人是否入队。曲棍球杆恒等式和 Vandermonde 恒等式 都可用分类选取证明。

Use / 怎么用

组合恒等式题优先找“同一对象两种计数”。If algebra is requested, use binomial theorem.

模板：组合证明 / Combinatorial Proof

写法：说明左边和右边都在计数同一个集合，只是分类方式不同。

深讲：Vandermonde 恒等式和 Hockey-Stick / More Binomial Identities

Vandermonde identity / Vandermonde 恒等式

$$
\sum_{k=0}^{r}\binom{m}{k}\binom{n}{r-k} = \binom{m+n}{r}. \tag{164}
$$

Combinatorial proof / 组合证明

从 $m+n$ 个人中选 r 人。把人分成两组，第一组 m 人，第二组 n 人。若从第一组选 k 人，则第二组必须选 $r-k$ 人。对所有可能的 k 求和，就得到左边。直接从全部 $m+n$ 人中选 r 人得到右边。

曲棍球杆恒等式 / Hockey-stick identity

$$
\sum_{k=r}^{n}\binom{k}{r} = \binom{n+1}{r+1}. \tag{165}
$$

Combinatorial proof / 组合证明

从集合 $\{0,1,2,\dots,n\}$ 中选 $r+1$ 个数。按最大元素 k 分类。若最大元素是 k，则其余 r 个必须从 $0,1,\dots,k-1$ 中选，有 $\binom{k}{r}$ 种。对 $k=r$ 到 n 求和得到左边。直接选 $r+1$ 个数得到右边。

How to use / 怎么用

看到卷积形式 $\sum\binom{m}{k}\binom{n}{r-k}$，想到 Vandermonde。看到连续相加 $\binom{r}{r}+\binom{r+1}{r}+\cdots$，想到 hockey-stick。

6.4 二项式定理与多项式定理 / Binomial and Multinomial Theorems

Binomial theorem / 二项式定理

$$
(x+y)^n = \sum_{k=0}^{n}\binom{n}{k}x^{n-k}y^k. \tag{166}
$$

Proof / 证明

展开 $(x+y)^n$ 时，每个因子选择 x 或 y。要得到 $x^{n-k}y^k$，需要从 n 个因子中选 k 个贡献 y，所以系数为 $\binom{n}{k}$。

Multinomial theorem / 多项式定理

$$(x_1+x_2+\cdots+x_m)^n=\sum_{k_1+\cdots+k_m=n}\frac{n!}{k_1!k_2!\cdots k_m!}\prod_{i=1}^{m}x_i^{k_i}. \tag{167}$$

Use / 怎么用

求展开式中特定项系数；把组合数求和转成取 $x=1,y=1$ 或 $x=1,y=-1$。

深讲：多项式定理和系数提取 / Multinomial Coefficients

Theorem / 多项式定理

$$(x_1+x_2+\cdots+x_m)^n=\sum_{k_1+\cdots+k_m=n}\frac{n!}{k_1!k_2!\cdots k_m!}x_1^{k_1}x_2^{k_2}\cdots x_m^{k_m}. \tag{168}$$

Why coefficient is multinomial / 系数为什么这样

展开时有 n 个括号。要得到 $x_1^{k_1}\cdots x_m^{k_m}$，必须在 k_1 个括号中选 x_1，在 k_2 个括号中选 x_2，依此类推。选择位置数为：

$$\binom{n}{k_1}\binom{n-k_1}{k_2}\cdots\binom{k_m}{k_m}=\frac{n!}{k_1!k_2!\cdots k_m!}. \tag{169}$$

How to use / 怎么用

求 $(x+y+z)^{10}$ 中 $x^2y^3z^5$ 的系数，直接用：

$$\frac{10!}{2!3!5!}. \tag{170}$$

6.5 鸽巢原理 / Pigeonhole Principle

Theorem / 基本鸽巢原理

若把 $n+1$ 个对象放入 n 个盒子，则至少一个盒子含有至少两个对象。

Generalized theorem / 广义鸽巢原理

若把 N 个对象放入 k 个盒子，则至少一个盒子含有至少：

$$\left\lceil\frac{N}{k}\right\rceil \tag{171}$$

个对象。

Proof / 证明

若每个盒子最多有 $\lceil N/k\rceil-1$ 个对象，则总数最多为 $k(\lceil N/k\rceil-1)<N$，矛盾。

Use / 怎么用

关键是设计“对象”和“盒子”。题中出现“至少两个相同”“必有一对”“保证至少”时优先考虑鸽巢原理。

6.6 星棒法 / Stars and Bars

Theorem / 非负整数解数

方程

$$x_1+x_2+\cdots+x_k=n,\qquad x_i\geq 0 \tag{172}$$

的非负整数解个数为：

$$\binom{n+k-1}{k-1}. \tag{173}$$

Proof / 证明

把 n 个星星排成一行，用 $k-1$ 根隔板分成 k 组。总共有 $n+k-1$ 个位置，从中选 $k-1$ 个放隔板。

Positive version / 正整数版本

若 $x_i \geq 1$，令 $y_i = x_i - 1 \geq 0$，得：

$$\binom{n-1}{k-1}. \tag{174}$$

Use / 怎么用

可重复组合、分配相同物品、整数解题都用星棒法。若有上界，常与容斥结合。

课堂展开：星棒法加上界 / Stars and Bars with Upper Bounds

Example / 例题

求非负整数解数量：

$$x_1 + x_2 + x_3 = 10, \qquad 0 \leq x_i \leq 5. \tag{175}$$

Solution / 解法

若没有上界，解数为：

$$\binom{10+3-1}{3-1} = \binom{12}{2} = 66. \tag{176}$$

现在排除坏事件 A_i: $x_i \geq 6$。若 $x_1 \geq 6$，令 $y_1 = x_1 - 6 \geq 0$，则：

$$y_1 + x_2 + x_3 = 4, \tag{177}$$

解数为：

$$\binom{4+3-1}{2} = \binom{6}{2} = 15. \tag{178}$$

三个变量对称，所以单个坏事件总贡献 $3 \cdot 15 = 45$。两个变量同时至少 6 不可能，因为总和只有 10。所以答案：

$$66 - 45 = 21. \tag{179}$$

How to use / 怎么用

有上界时先忽略上界，再用容斥减去超过上界的解。

6.7 有重复元素的排列与多重集合 / Repetition and Multisets

Formula / 公式

若多重集合中共有 n 个元素，其中第 i 类重复 n_i 次，且 $n_1 + \cdots + n_k = n$，不同排列数为：

$$\frac{n!}{n_1! n_2! \cdots n_k!}. \tag{180}$$

Proof / 证明

先把所有重复元素暂时标号，有 $n!$ 种排列。每类内部的 $n_i!$ 个标号排列不改变最终序列，所以除以这些重复数。

Use / 怎么用

字母排列、密码字符串、路径中固定数量的右/上步数都用这个公式。

补充：广义排列组合 / Generalized Permutations and Combinations

Sequences with repetition / 可重复序列

从 n 个元素中选长度为 r 的有序序列，允许重复，数量是：

$$n^r. \tag{181}$$

Proof / 证明

每个位置都有 n 种选择，共 r 个位置，由乘法原则得 n^r。

Combinations with repetition / 可重复组合

从 n 类对象中选 r 个，允许重复且不看顺序，数量为：

$$\binom{n+r-1}{r} = \binom{n+r-1}{n-1}. \tag{182}$$

Proof / 证明

设 x_i 是第 i 类对象选了多少个，则：

$$x_1 + x_2 + \cdots + x_n = r, \qquad x_i \geq 0. \tag{183}$$

由星棒法，解数为 $\binom{r+n-1}{n-1}$。

Circular permutations / 圆排列

n 个不同对象围成一圈，若旋转视为相同，则不同圆排列数为：

$$(n-1)!. \tag{184}$$

Proof / 证明

普通排列有 $n!$ 种。每个圆排列被 n 个线性排列表示，因为可以从圈上任意一个位置切开作为开头。所以数量为 $n!/n = (n-1)!$。

Distribution into boxes / 分配到盒子

r 个不同球放入 n 个不同盒子，允许空盒，数量为：

$$n^r. \tag{185}$$

r 个相同球放入 n 个不同盒子，允许空盒，数量为：

$$\binom{r+n-1}{n-1}. \tag{186}$$

6.8 错排 / Derangements

Theorem / 错排公式

n 个元素的错排数为：

$$D_n = n! \sum_{k=0}^{n} \frac{(-1)^k}{k!}. \tag{187}$$

递推式：

$$D_n = (n-1)(D_{n-1} + D_{n-2}), \qquad D_0 = 1, \quad D_1 = 0. \tag{188}$$

Proof / 容斥证明

令 A_i 为第 i 个元素固定不动的排列集合。错排数是没有任何 A_i 发生的排列数：

$$D_n = n! - \binom{n}{1}(n-1)! + \binom{n}{2}(n-2)! - \cdots + (-1)^n \binom{n}{n}0!. \tag{189}$$

化简得到公式。

Use / 怎么用

“每个人都拿不到自己的物品”“没有位置保持原样”就是错排。

课堂展开：错排与容斥 / Derangements by Inclusion-Exclusion

Example / 例题

四个人随机拿四顶帽子，没人拿到自己帽子的方式有多少?

Solution / 解法

这是 D_4：

$$D_4 = 4!\left(1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} + \frac{1}{4!}\right). \tag{190}$$

计算：

$$D_4 = 24\left(1 - 1 + \frac{1}{2} - \frac{1}{6} + \frac{1}{24}\right) = 24 \cdot \frac{9}{24} = 9. \tag{191}$$

Interpretation / 解释

总排列 24 个。减去至少一个人拿对的排列，加回至少两个人拿对的排列，再交替处理。

6.9 满射函数与 Stirling 数 / Onto Functions and Stirling Numbers

Formula / 满射函数数量

从 n 元集合到 k 元集合的满射数量为：

$$\sum_{i=0}^{k} (-1)^i \binom{k}{i} (k-i)^n. \tag{192}$$

Proof / 证明

总函数数为 k^n。令 A_j 表示陪域第 j 个元素没有被命中。用容斥排除这些坏事件。

Stirling numbers / 第二类 Stirling 数

$S(n,k)$ 表示把 n 个有标号元素分成 k 个非空无标号块的数量。

$$S(n,k) = S(n-1,k-1) + kS(n-1,k). \tag{193}$$

Proof / 证明

看第 n 个元素：它单独成一块，有 $S(n-1,k-1)$ 种；或者加入已有 k 块之一，有 $kS(n-1,k)$ 种。

Use / 怎么用

满射数还等于 $k!S(n,k)$，因为先把定义域分成 k 个非空块，再把块标号到陪域元素。

深讲：满射计数和第二类 Stirling 数 / Onto Functions and Stirling Numbers

Onto function formula / 满射公式

从 n 元集合到 k 元集合的满射数：

$$\sum_{i=0}^{k}(-1)^i\binom{k}{i}(k-i)^n. \tag{194}$$

Detailed proof / 详细证明

所有函数共有 k^n 个。令 A_j 表示陪域中第 j 个元素没有被任何输入映到。满射就是没有任何 A_j 发生。

如果指定 i 个陪域元素没有被用到，那么每个输入只能映到剩下的 $k-i$ 个元素，所以有 $(k-i)^n$ 个函数。选择这 i 个没用到的元素有 $\binom{k}{i}$ 种。由容斥：

$$\#\text{onto} = \binom{k}{0}k^n - \binom{k}{1}(k-1)^n + \binom{k}{2}(k-2)^n - \cdots. \tag{195}$$

Connection to Stirling numbers / 与 Stirling 数的关系

$S(n,k)$ 把 n 个有标号元素分成 k 个非空无标号块。若再给这 k 个块贴上陪域的 k 个标签，就得到满射。因此：

$$\#\text{满射函数} = k!S(n,k). \tag{196}$$

所以：

$$S(n,k) = \frac{1}{k!}\sum_{i=0}^{k}(-1)^i\binom{k}{i}(k-i)^n. \tag{197}$$

6.10 Ramsey 例子 / Ramsey Example

Theorem / 定理

$$R(3,3) = 6. \tag{198}$$

也就是说，任意 6 个人中，总有三个人彼此认识，或三个人彼此不认识。

Proof / 证明

把人看成完全图 K_6 的顶点，边用红色表示认识，蓝色表示不认识。固定顶点 v，它有 5 条边。由鸽巢原理，至少有 3 条同色。设 v 连到 a,b,c 的边都红。若 a,b,c 之间有红边，则与 v 形成红三角形；若没有红边，则 a,b,c 之间全是蓝边，形成蓝三角形。下界 $R(3,3) > 5$ 可用五边形红边和对角线蓝边构造无单色三角形的 K_5。

Use / 怎么用

Ramsey 题的典型策略是固定一个点，然后对相邻边颜色用鸽巢原理。

6.11 Catalan 数 / Catalan Numbers

Definition / 定义

Catalan 数为：

$$C_n = \frac{1}{n+1}\binom{2n}{n}. \tag{199}$$

等价形式：

$$C_n = \binom{2n}{n} - \binom{2n}{n+1}. \tag{200}$$

Common objects counted / 常见计数对象

C_n 计数很多等价对象：

- n 对括号的合法匹配方式。
- 从 $(0,0)$ 到 (n,n)、不越过对角线 $y=x$ 的格路径数。
- $n+1$ 个叶子的满二叉树形状数。
- 凸 $(n+2)$ 边形的三角剖分数。

Proof idea / 反射法证明思路

从 $(0,0)$ 到 (n,n) 的所有路径有 $\binom{2n}{n}$ 条。坏路径第一次越过对角线后反射，可与从 $(0,0)$ 到 $(n-1,n+1)$ 的路径建立双射，坏路径数为 $\binom{2n}{n+1}$。所以好路径数为：

$$\binom{2n}{n}-\binom{2n}{n+1}=\frac{1}{n+1}\binom{2n}{n}. \tag{201}$$

Use / 怎么用

看到“合法括号”“不越过对角线”“二叉树形状”“多边形三角剖分”，先想到 Catalan 数。

7. 离散概率 / Discrete Probability

7.1 概率空间 / Probability Spaces

Definition / 定义

概率空间由样本空间 S 和概率函数 $\Pr$ 组成，满足：

$$\Pr(S)=1,\qquad \Pr(A)\geq 0,\qquad A\cap B=\varnothing \Rightarrow \Pr(A\cup B)=\Pr(A)+\Pr(B). \tag{202}$$

若所有结果等可能，则：

$$\Pr(A)=\frac{|A|}{|S|}. \tag{203}$$

Use / 怎么用

概率题第一步必须写清样本空间。等可能时用计数；不等可能时不能直接数个数。

模板：概率四步法 / Probability Four-Step Method

1. 写出样本空间 S。
2. 判断是否等可能。
3. 定义事件 A。
4. 用计数、条件概率、独立性或补事件计算。

7.2 概率规则 / Probability Rules

Formulas / 公式

$$\begin{aligned}\Pr(A^c)&=1-\Pr(A),\\ \Pr(A\cup B)&=\Pr(A)+\Pr(B)-\Pr(A\cap B),\\ \Pr(A-B)&=\Pr(A)-\Pr(A\cap B),\\ \Pr(A\cup B)&\leq \Pr(A)+\Pr(B).\end{aligned} \tag{204}$$

最后一条是 union bound / 并合界。

Proof / 证明

$A \cup B$ 被 A 和 B 的概率相加时，$A \cap B$ 被加了两次，所以减去一次。并合界来自减去的交集概率非负。

Use / 怎么用

“至少一个事件发生”可用补事件或并合界。精确值用容斥，上界用 union bound。

7.3 条件概率与贝叶斯 / Conditional Probability and Bayes

Definitions / 定义

$$\Pr(A \mid B) = \frac{\Pr(A \cap B)}{\Pr(B)} \qquad (\Pr(B) > 0). \tag{205}$$

乘法公式：

$$\Pr(A \cap B) = \Pr(A \mid B)\Pr(B) = \Pr(B \mid A)\Pr(A). \tag{206}$$

全概率公式：若 $B_1, \dots, B_n$ 构成样本空间划分，则：

$$\Pr(A) = \sum_{i=1}^{n} \Pr(A \mid B_i)\Pr(B_i). \tag{207}$$

Bayes 公式：

$$\Pr(B_j \mid A) = \frac{\Pr(A \mid B_j)\Pr(B_j)}{\sum_{i=1}^{n} \Pr(A \mid B_i)\Pr(B_i)}. \tag{208}$$

Proof / 证明

条件概率定义给出乘法公式；把 A 分解成互不相交的 $A \cap B_i$ 得全概率公式；Bayes 由 $\Pr(B_j \mid A) = \Pr(A \cap B_j)/\Pr(A)$ 和全概率公式得到。

Use / 怎么用

看到“已知检测阳性，患病概率是多少”这类反向条件概率，必须用 Bayes，不能把 $\Pr(A \mid B)$ 和 $\Pr(B \mid A)$ 混用。

课堂展开：概率：条件概率和 Bayes 的完整表格法 / Bayes with a Table

Example / 例题

某病患病率为 1%。检测对患病者阳性率为 99%，对未患病者假阳性率为 5%。若某人检测阳性，患病概率是多少?

Solution / 解法

设 D 表示患病，$+$ 表示阳性。

已知：

$$\Pr(D) = 0.01, \quad \Pr(D^c) = 0.99, \quad \Pr(+ \mid D) = 0.99, \quad \Pr(+ \mid D^c) = 0.05. \tag{209}$$

由 Bayes 公式：

$$\Pr(D \mid +) = \frac{\Pr(+ \mid D)\Pr(D)}{\Pr(+ \mid D)\Pr(D) + \Pr(+ \mid D^c)\Pr(D^c)}. \tag{210}$$

代入：

$$\Pr(D \mid +) = \frac{0.99 \cdot 0.01}{0.99 \cdot 0.01 + 0.05 \cdot 0.99} = \frac{0.0099}{0.0594} = \frac{1}{6}. \tag{211}$$

How to use / 怎么用

基础率 / base rate 很重要。即使检测很准，若疾病很罕见，阳性中仍可能有很多假阳性。

深讲：条件独立和全概率 / Conditional Independence and Total Probability

Partition / 划分

事件 $B_1, \dots, B_n$ 构成样本空间划分，指它们两两互斥且并为整个样本空间：

$$B_i \cap B_j = \varnothing \quad (i \neq j), \qquad \bigcup_{i=1}^{n} B_i = S. \tag{212}$$

Why total probability holds / 全概率为什么成立

事件 A 可拆成互不相交的部分：

$$A = (A \cap B_1) \cup \cdots \cup (A \cap B_n). \tag{213}$$

因此：

$$\Pr(A) = \sum_{i=1}^{n} \Pr(A \cap B_i) = \sum_{i=1}^{n} \Pr(A \mid B_i) \Pr(B_i). \tag{214}$$

Conditional independence / 条件独立

A 与 B 在给定 C 下条件独立，指：

$$\Pr(A \cap B \mid C) = \Pr(A \mid C) \Pr(B \mid C). \tag{215}$$

这不等同于普通独立。普通独立不一定推出条件独立，条件独立也不一定推出普通独立。

How to use / 怎么用

Bayes 题最好先找一个划分，例如“患病/未患病”“来自机器 1/机器 2/机器 3”。分母通常就是全概率公式。

7.4 独立性 / Independence

Definition / 定义

事件 A 与 B 独立当且仅当：

$$\Pr(A \cap B) = \Pr(A) \Pr(B). \tag{216}$$

若 $\Pr(B) > 0$，等价于：

$$\Pr(A \mid B) = \Pr(A). \tag{217}$$

多个事件相互独立要求任意非空子集都满足乘法公式：

$$\Pr\left(\bigcap_{i \in I} A_i\right) = \prod_{i \in I} \Pr(A_i). \tag{218}$$

Proof / 证明

二事件等价式直接由条件概率定义除以 $\Pr(B)$ 得到。多事件独立是定义，不能只检查两两独立。

Use / 怎么用

独立时可以相乘；互斥时交集为 0。除非某事件概率为 0，互斥和独立通常不能同时成立。

7.5 随机变量、期望、方差 / Random Variables, Expectation, Variance

Definitions / 定义

离散随机变量 X 的期望：

$$\mathbb{E}[X] = \sum_{x} x \Pr(X = x). \tag{219}$$

若 X 是样本空间上的函数，也可写：

$$\mathbb{E}[X] = \sum_{s \in S} X(s) \Pr(s). \tag{220}$$

线性性：

$$\mathbb{E}[aX + bY + c] = a\mathbb{E}[X] + b\mathbb{E}[Y] + c. \tag{221}$$

方差：

$$\mathrm{Var}(X) = \mathbb{E}\big[(X - \mathbb{E}[X])^2\big] = \mathbb{E}[X^2] - \mathbb{E}[X]^2. \tag{222}$$

若 X, Y 独立：

$$\mathrm{Var}(X + Y) = \mathrm{Var}(X) + \mathrm{Var}(Y). \tag{223}$$

Proof / 证明

期望线性性由求和分配律得到，不需要独立。方差公式展开平方：

$$\mathbb{E}[X^2 - 2X\mathbb{E}[X] + \mathbb{E}[X]^2] = \mathbb{E}[X^2] - \mathbb{E}[X]^2. \tag{224}$$

Use / 怎么用

计数随机对象时常用指示变量 I_i：若事件 A_i 发生，则 $I_i = 1$，否则 0。于是 $\mathbb{E}[I_i] = \Pr(A_i)$，总数 $X = \sum I_i$，所以期望可逐项相加。

课堂展开：期望：指示变量法 / Indicator Variables

Example / 例题

随机排列 n 个元素，期望有多少个 fixed points / 不动点？

Solution / 解法

令 I_i 表示第 i 个元素固定：

$$I_i = \begin{cases} 1, & \text{if element } i \text{ stays in position } i, \\ 0, & \text{otherwise.} \end{cases} \tag{225}$$

总不动点数：

$$X = I_1 + I_2 + \cdots + I_n. \tag{226}$$

对每个 i，第 i 个元素留在原位的概率是 $1/n$，所以：

$$\mathbb{E}[I_i] = \frac{1}{n}. \tag{227}$$

由期望线性性：

$$\mathbb{E}[X] = \sum_{i=1}^{n} \mathbb{E}[I_i] = n \cdot \frac{1}{n} = 1. \tag{228}$$

Key point / 关键点

这里不需要事件独立。期望线性性永远成立。

深讲：方差、协方差与独立性 / Variance, Covariance, Independence

Variance expansion / 方差展开

$$\mathrm{Var}(X) = \mathbb{E}[X^2] - \mathbb{E}[X]^2. \tag{229}$$

对和：

$$\mathrm{Var}(X+Y) = \mathrm{Var}(X) + \mathrm{Var}(Y) + 2\,\mathrm{Cov}(X,Y), \tag{230}$$

其中：

$$\mathrm{Cov}(X,Y) = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]. \tag{231}$$

若 X, Y 独立，则 $\mathbb{E}[XY] = \mathbb{E}[X]\mathbb{E}[Y]$，所以协方差为 0。

Important warning / 重要警告

协方差为 0 不一定推出独立。独立比不相关更强。

How to use / 怎么用

如果随机变量不是独立的，不能直接把方差相加。期望可以直接相加，方差不可以。

7.6 常见分布 / Common Distributions

Bernoulli / 伯努利分布

$$\Pr(X=1) = p, \qquad \mathbb{E}[X] = p, \qquad \mathrm{Var}(X) = p(1-p). \tag{232}$$

Binomial / 二项分布

若 $X \sim \mathrm{Bin}(n,p)$，则：

$$\Pr(X=k) = \binom{n}{k} p^k (1-p)^{n-k}, \qquad \mathbb{E}[X] = np, \qquad \mathrm{Var}(X) = np(1-p). \tag{233}$$

Geometric / 几何分布

若 X 是首次成功所需试验次数，则：

$$\Pr(X=k) = (1-p)^{k-1} p, \qquad \mathbb{E}[X] = \frac{1}{p}. \tag{234}$$

Poisson / 泊松分布

$$\Pr(X=k) = e^{-\lambda} \frac{\lambda^k}{k!}, \qquad \mathbb{E}[X] = \lambda, \qquad \mathrm{Var}(X) = \lambda. \tag{235}$$

Use / 怎么用

固定次数数成功用二项；等首次成功用几何；稀有独立事件近似用 Poisson。

深讲：概率分布公式的来源 / Where Distribution Formulas Come From

Binomial distribution / 二项分布

$X \sim \mathrm{Bin}(n,p)$ 表示 n 次独立试验中成功次数。要让 $X=k$，需要选出哪 k 次成功，有 $\binom{n}{k}$ 种；每种具体模式的概率是：

$$p^k (1-p)^{n-k}. \tag{236}$$

所以：

$$\Pr(X=k)=\binom{n}{k}p^k(1-p)^{n-k}. \tag{237}$$

期望可用指示变量：令 I_i 表示第 i 次成功，则 $X=I_1+\cdots+I_n$，所以：

$$\mathbb{E}[X]=np. \tag{238}$$

方差因为独立可加：

$$\mathrm{Var}(X)=np(1-p). \tag{239}$$

Geometric distribution / 几何分布

首次成功在第 k 次，意味着前 $k-1$ 次失败，第 k 次成功：

$$\Pr(X=k)=(1-p)^{k-1}p. \tag{240}$$

期望可用递推。设 $E=\mathbb{E}[X]$。第一次试验消耗 1 次；若失败，概率 $1-p$，还要重新等待 E 次：

$$E=1+(1-p)E. \tag{241}$$

解得：

$$E=\frac{1}{p}. \tag{242}$$

7.7 偏差界 / Deviation Bounds

Markov inequality / Markov 不等式

若 $X\geq 0$，则：

$$\Pr(X\geq a)\leq\frac{\mathbb{E}[X]}{a}. \tag{243}$$

Proof / 证明

因为 $X\geq a$ 的部分至少贡献 $a\Pr(X\geq a)$ 给期望，所以 $\mathbb{E}[X]\geq a\Pr(X\geq a)$。

Chebyshev inequality / Chebyshev 不等式

$$\Pr(|X-\mathbb{E}[X]|\geq a)\leq\frac{\mathrm{Var}(X)}{a^2}. \tag{244}$$

Proof / 证明

对非负随机变量 $(X-\mathbb{E}[X])^2$ 使用 Markov 不等式。

Chernoff bound / Chernoff 界

若 X 是独立 Bernoulli 变量之和，$\mu=\mathbb{E}[X]$，则常用形式为：

$$\Pr(X\geq(1+\delta)\mu)\leq\left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mu}\quad(\delta>0). \tag{245}$$

Use / 怎么用

Markov 只需非负，最粗；Chebyshev 需要方差；Chernoff 需要独立 Bernoulli 和，最强。

深讲：偏差界怎么选 / Choosing Markov, Chebyshev, Chernoff

Markov / Markov 不等式

只要求 $X \geq 0$，所以适用范围最大，但界最粗：

$$\Pr(X \geq a) \leq \frac{\mathbb{E}[X]}{a}. \tag{246}$$

Chebyshev / Chebyshev 不等式

需要知道方差，适合控制偏离均值：

$$\Pr(|X - \mu| \geq k\sigma) \leq \frac{1}{k^2}. \tag{247}$$

Chernoff / Chernoff 界

通常用于独立 0-1 随机变量之和。它给指数级小的尾概率。

Example / 例子

若 X 是 1000 次公平投硬币中正面次数，则 $\mu = 500$。想估计 $X \geq 600$，这是独立 Bernoulli 和的尾部，Chernoff 比 Chebyshev 更合适。

7.8 生日问题 / Birthday Principle

Formula / 公式

k 个人生日都不同的概率近似为：

$$\prod_{i=0}^{k-1} \left(1 - \frac{i}{365}\right) \approx e^{-k(k-1)/(2\cdot 365)}. \tag{248}$$

至少两人生日相同的概率为：

$$1 - rac365 \cdot 364 \cdots (365 - k + 1)365^k. \tag{249}$$

Proof / 证明

逐个安排生日，第 $i+1$ 个人要避开前面 i 个生日，概率为 $1 - i/365$。近似使用 $1 - x \approx e^{-x}$。

Use / 怎么用

哈希碰撞、随机抽样冲突、密码 nonce 碰撞常用生日近似。

8. 高级计数、递推与生成函数 / Advanced Counting, Recurrences, and Generating Functions

8.1 递推关系 / Recurrence Relations

Definition / 定义

递推关系用前面若干项定义后续项。例如：

$$a_n = 3a_{n-1} + 2, \qquad a_0 = 1. \tag{250}$$

Use / 怎么用

解递推常见方法：迭代展开、特征方程、生成函数、递归树、主定理。

8.2 一阶线性递推 / First-Order Linear Recurrences

Theorem / 定理

若

$$a_n = ca_{n-1} + d, \tag{251}$$

则当 $c \neq 1$ 时：

$$a_n = c^n a_0 + d\frac{c^n - 1}{c - 1}. \tag{252}$$

当 $c = 1$ 时：

$$a_n = a_0 + nd. \tag{253}$$

Proof / 证明

迭代展开：

$$a_n = c^n a_0 + d(1 + c + c^2 + \cdots + c^{n-1}). \tag{254}$$

再使用等比求和公式。

Use / 怎么用

遇到每步乘固定常数再加固定项的递推，直接套用。算法中“规模减一，每层固定代价”也常化成这种形式。

8.3 常系数齐次线性递推 / Linear Homogeneous Recurrences

Theorem / 二阶不同根

若

$$a_n = c_1 a_{n-1} + c_2 a_{n-2}, \tag{255}$$

特征方程为：

$$r^2 - c_1 r - c_2 = 0. \tag{256}$$

若有不同根 r_1, r_2，则：

$$a_n = \alpha r_1^n + \beta r_2^n. \tag{257}$$

若有重根 r，则：

$$a_n = (\alpha + \beta n) r^n. \tag{258}$$

Proof / 证明

尝试解 $a_n = r^n$，代入递推得到特征方程。不同根线性组合仍满足线性齐次递推。常数 α, β 由初值确定。重根情况需要第二个线性独立解 nr^n。

Use / 怎么用

Fibonacci 数满足 $F_n = F_{n-1} + F_{n-2}$，特征根为：

$$\phi = \frac{1 + \sqrt{5}}{2}, \qquad \psi = \frac{1 - \sqrt{5}}{2}. \tag{259}$$

因此：

$$F_n = \frac{\phi^n - \psi^n}{\sqrt{5}}. \tag{260}$$

课堂展开：递推：特征方程完整算例 / Characteristic Equation Example

Example / 例题

解递推：

$$a_n = 5a_{n-1} - 6a_{n-2}, \qquad a_0 = 2, \quad a_1 = 5. \tag{261}$$

Solution / 解法

设 $a_n = r^n$，代入：

$$r^n = 5r^{n-1} - 6r^{n-2}. \tag{262}$$

除以 r^{n-2}：

$$r^2 = 5r - 6. \tag{263}$$

特征方程为：

$$r^2 - 5r + 6 = 0 = (r-2)(r-3). \tag{264}$$

所以通解：

$$a_n = \alpha 2^n + \beta 3^n. \tag{265}$$

用初值：

$$\begin{aligned} a_0 = 2: &\quad \alpha + \beta = 2, \\ a_1 = 5: &\quad 2\alpha + 3\beta = 5. \end{aligned} \tag{266}$$

解得 $\beta = 1$，$\alpha = 1$。因此：

$$a_n = 2^n + 3^n. \tag{267}$$

How to use / 怎么用

常系数齐次线性递推的流程固定：写特征方程，求根，写通解，用初值解常数。

8.4 非齐次递推 / Nonhomogeneous Recurrences

Method / 方法

非齐次递推：

$$a_n - c_1 a_{n-1} - \cdots - c_k a_{n-k} = F(n). \tag{268}$$

通解为：

$$a_n = a_n^{(h)} + a_n^{(p)}, \tag{269}$$

其中 $a_n^{(h)}$ 是齐次解，$a_n^{(p)}$ 是一个特解。

Use / 怎么用

若 $F(n)$ 是多项式、指数、或它们乘积，猜同类型特解；若与齐次解重复，要乘以足够高次的 n。

深讲：非齐次递推：猜特解的规则 / Nonhomogeneous Recurrences

General form / 一般形式

$$a_n - c_1 a_{n-1} - \cdots - c_k a_{n-k} = F(n). \tag{270}$$

先解齐次部分，再找一个特解。

Example / 例题

解：

$$a_n = 2a_{n-1} + 3, \qquad a_0 = 1. \tag{271}$$

齐次解满足 $a_n^{(h)} = 2a_{n-1}^{(h)}$，所以：

$$a_n^{(h)} = C2^n. \tag{272}$$

因为右边非齐次项是常数 3，猜特解 $a_n^{(p)} = A$。代入：

$$A = 2A + 3, \tag{273}$$

所以 $A = -3$。通解：

$$a_n = C2^n - 3. \tag{274}$$

用 $a_0 = 1$ 得 $C - 3 = 1$，所以 $C = 4$。因此：

$$a_n = 4 \cdot 2^n - 3. \tag{275}$$

Repeated guess issue / 重复时怎么办

若猜的特解形式已经出现在齐次解中，要乘以 n。例如 $a_n = 2a_{n-1} + 2^n$，因为 2^n 是齐次解的一部分，特解应猜 $An2^n$。

8.5 分治递推与主定理 / Divide-and-Conquer and Master Theorem

Master theorem / 主定理

对

$$T(n) = aT\left(\frac{n}{b}\right) + f(n), \qquad a \geq 1, \quad b > 1, \tag{276}$$

令 $d = \log_b a$。

- 若 $f(n) = O(n^{d-\epsilon})$，则 $T(n) = \Theta(n^d)$。
- 若 $f(n) = \Theta(n^d \log^k n)$，则 $T(n) = \Theta(n^d \log^{k+1} n)$。
- 若 $f(n) = \Omega(n^{d+\epsilon})$ 且满足 正则性条件 $af(n/b) \leq cf(n)$ for some $c < 1$，则 $T(n) = \Theta(f(n))$。

Proof idea / 证明思路

递归树第 i 层有 a^i 个子问题，每个规模 n/b^i，该层总代价为 $a^i f(n/b^i)$。比较根部代价、叶子代价、每层均衡时的总和。

Use / 怎么用

先算 $n^{\log_b a}$，再和 $f(n)$ 比。Merge sort: $T(n) = 2T(n/2) + n$，因为 $n^{\log_2 2} = n$，所以 $T(n) = \Theta(n \log n)$。

课堂展开：主定理：三种情况怎么判断 / Master Theorem Examples

Example 1 / 平衡情况

$$T(n) = 2T(n/2) + n. \tag{277}$$

这里 $a = 2$，$b = 2$，所以：

$$n^{\log_b a} = n^{\log_2 2} = n. \tag{278}$$

$f(n) = n$ 与 $n^{\log_b a}$ 同阶，所以：

$$T(n) = \Theta(n \log n). \tag{279}$$

Example 2 / 叶子主导

$$T(n) = 4T(n/2) + n. \tag{280}$$

这里：

$$n^{\log_2 4} = n^2. \tag{281}$$

$f(n) = n = O(n^{2-1})$，所以叶子层主导：

$$T(n) = \Theta(n^2). \tag{282}$$

Example 3 / 根部主导

$$T(n) = 2T(n/2) + n^2. \tag{283}$$

这里 $n^{\log_2 2} = n$，而 $f(n) = n^2$ 更大，并满足 正则性条件，所以：

$$T(n) = \Theta(n^2). \tag{284}$$

How to use / 怎么用

主定理不要死背文字，先算分界函数 $n^{\log_b a}$，再比较 $f(n)$ 比它小、相等还是大。

8.6 生成函数 / Generating Functions

Definition / 定义

序列 $a_0, a_1, a_2, \ldots$ 的普通生成函数为：

$$A(x) = \sum_{n=0}^{\infty} a_n x^n. \tag{285}$$

Basic generating functions / 基本生成函数

$$\begin{aligned}
\frac{1}{1-x} &= \sum_{n=0}^{\infty} x^n, \\
\frac{1}{(1-x)^2} &= \sum_{n=0}^{\infty} (n+1)x^n, \\
\frac{1}{(1-x)^k} &= \sum_{n=0}^{\infty} \binom{n+k-1}{k-1} x^n, \\
(1+x)^n &= \sum_{k=0}^{n} \binom{n}{k} x^k.
\end{aligned} \tag{286}$$

Proof / 证明

第一条是无限等比级数。第二条可对第一条求导后乘以 x 或直接求导得到。第三条对应星棒法或广义二项式定理。

Use / 怎么用

生成函数把“选取数量”变成“系数提取”。例如 $\prod_i (1 + x^{w_i})$ 中 x^n 的系数表示用各物品至多一次凑出重量 n 的方法数。

课堂展开：生成函数：把限制翻译成因子 / Generating Functions as Translators

Example / 例题

用生成函数求方程

$$x_1 + x_2 + x_3 = 8, \qquad x_1 \geq 0, \quad x_2 \in \{0, 1, 2\}, \quad x_3 \text{ is even} \tag{287}$$

的解数。

Solution / 解法

每个变量对应一个生成函数因子。

$x_1 \ge 0$：

$$1 + x + x^2 + x^3 + \cdots = \frac{1}{1-x}. \tag{288}$$

$x_2 \in \{0,1,2\}$：

$$1 + x + x^2. \tag{289}$$

x_3 为非负偶数：

$$1 + x^2 + x^4 + x^6 + \cdots = \frac{1}{1-x^2}. \tag{290}$$

总生成函数：

$$G(x) = \frac{1}{1-x}(1 + x + x^2)\frac{1}{1-x^2}. \tag{291}$$

答案是 x^8 的系数。也可以直接枚举 x_2：

- 若 $x_2 = 0$，则 $x_1 + x_3 = 8$，x_3 可为 $0, 2, 4, 6, 8$，有 5 种。
- 若 $x_2 = 1$，则 $x_1 + x_3 = 7$，x_3 可为 $0, 2, 4, 6$，有 4 种。
- 若 $x_2 = 2$，则 $x_1 + x_3 = 6$，x_3 可为 $0, 2, 4, 6$，有 4 种。

总数为：

$$5 + 4 + 4 = 13. \tag{292}$$

How to use / 怎么用

生成函数最适合“每个变量有不同限制”的计数题。每个限制先翻译成允许指数集合，再相乘取目标系数。

深讲：生成函数解递推 / Solving Recurrences by Generating Functions

Example / Fibonacci generating function

令：

$$F_0 = 0, \quad F_1 = 1, \quad F_n = F_{n-1} + F_{n-2}. \tag{293}$$

设生成函数：

$$F(x) = \sum_{n=0}^{\infty} F_n x^n. \tag{294}$$

递推对 $n \ge 2$ 成立。考虑：

$$F(x) - x = \sum_{n=2}^{\infty} F_n x^n. \tag{295}$$

由递推：

$$\sum_{n=2}^{\infty} F_n x^n = \sum_{n=2}^{\infty} F_{n-1} x^n + \sum_{n=2}^{\infty} F_{n-2} x^n. \tag{296}$$

右边分别为：

$$xF(x), \qquad x^2 F(x). \tag{297}$$

所以：

$$F(x) - x = xF(x) + x^2F(x). \tag{298}$$

解得：

$$F(x) = \frac{x}{1 - x - x^2}. \tag{299}$$

How to use / 怎么用

生成函数把递推变成代数方程。步骤：定义 $A(x)$，把递推乘 x^n 后求和，处理初值，解出 $A(x)$。

8.7 双重求和与换序 / Double Sums and Changing Order

Formulas / 公式

$$\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij} = \sum_{j=1}^{n}\sum_{i=1}^{m} a_{ij}. \tag{300}$$

三角区域常见换序：

$$\sum_{i=1}^{n}\sum_{j=1}^{i} a_{ij} = \sum_{j=1}^{n}\sum_{i=j}^{n} a_{ij}. \tag{301}$$

Proof / 证明

两边都在对同一组指标对求和，只是顺序不同。有限求和可以任意重排。

Use / 怎么用

当内层和难算时换序。计数证明中可把求和看成“数格点”。

9. 关系、等价关系、偏序 / Relations, Equivalence Relations, and Partial Orders

9.1 关系及性质 / Relations and Properties

Definition / 定义

从 A 到 B 的关系是 $A \times B$ 的子集。集合 A 上的关系是 $A \times A$ 的子集。

Properties / 性质

- Reflexive / 自反: $\forall a \in A,\ aRa$.
- Irreflexive / 反自反: $\forall a \in A,\ \neg aRa$.
- Symmetric / 对称: $aRb \Rightarrow bRa$.
- Antisymmetric / 反对称: $(aRb \wedge bRa) \Rightarrow a = b$.
- Asymmetric / 非对称: $aRb \Rightarrow \neg bRa$.
- Transitive / 传递: $(aRb \wedge bRc) \Rightarrow aRc$.

Use / 怎么用

判断性质时，先写定义，再找反例。反对称不是“没有双向边”，而是只允许不同元素之间不能双向。

补充：n 元关系与数据库操作 / n-ary Relations and Databases

Definition / 定义

n 元关系是笛卡尔积 $A_1 \times A_2 \times \cdots \times A_n$ 的子集。数据库表可以看成 n 元关系，每一行是一个 n 元组。

Selection / 选择

选择操作按条件筛选行。例如只保留成绩大于 90 的记录。

Projection / 投影

投影操作保留若干列，丢弃其他列。若出现重复行，关系模型中通常只保留一份。

Join / 连接

连接操作把两个关系按共同属性匹配并合并行。

Use / 怎么用

Rosen 的 n 元关系主要是数据库建模。题目通常要求把表操作翻译成集合和关系操作。

9.2 关系表示与闭包 / Representations and Closures

Matrix representation / 矩阵表示

关系 R 的矩阵 M_R 定义为：

$$(M_R)_{ij} = 1 \iff a_i R a_j. \tag{302}$$

Closures / 闭包

- 自反闭包：加入所有 (a, a)。
- 对称闭包：若有 (a, b)，加入 (b, a)。
- 传递闭包：加入所有可由路径到达的 (a, b)。

Warshall algorithm idea / Warshall 思路

逐个允许中间点 $1, 2, \ldots, k$，更新可达性：

$$W_{ij}^{(k)} = W_{ij}^{(k-1)} \vee \left(W_{ik}^{(k-1)} \wedge W_{kj}^{(k-1)} \right). \tag{303}$$

Proof / 证明

从 i 到 j 且中间点只来自前 k 个顶点的路径，要么不用第 k 个顶点，要么经过 k，分成 $i \to k$ 和 $k \to j$ 两段。

Use / 怎么用

可达性、先修关系、依赖关系都可以用传递闭包。

补充：关系的复合、幂与可达性 / Composition and Powers of Relations

Composition / 关系复合

若 R 是从 A 到 B 的关系，S 是从 B 到 C 的关系，则复合关系 $S \circ R$ 定义为：

$$a (S \circ R) c \iff \exists b \in B \, (a R b \wedge b S c). \tag{304}$$

Powers / 关系幂

集合 A 上关系 R 的幂定义为：

$$R^1 = R, \qquad R^{n+1} = R^n \circ R. \tag{305}$$

Path interpretation / 路径解释

在有向图中，$(a, b) \in R^n$ 当且仅当存在从 a 到 b 的长度为 n 的有向走路。

Transitive closure / 传递闭包

若 $|A| = N$，则传递闭包可写成：

$$R^+ = R \cup R^2 \cup \cdots \cup R^N. \tag{306}$$

如果还要自反传递闭包，则：

$$R^* = I_A \cup R \cup R^2 \cup \cdots \cup R^N. \tag{307}$$

Use / 怎么用

关系幂把“多步到达”变成代数运算。Warshall 算法就是高效求传递闭包。

深讲：关系闭包：矩阵和图的视角 / Closures by Graphs and Matrices

Reflexive closure / 自反闭包

矩阵上就是把主对角线全改为 1。图上就是给每个顶点加自环。

Symmetric closure / 对称闭包

矩阵上就是把 M 改为：

$$M \vee M^T. \tag{308}$$

图上就是每条有向边都补上反向边。

Transitive closure / 传递闭包

传递闭包表示“可达性”。如果关系看成有向图，则 (a, b) 在传递闭包中，当且仅当存在从 a 到 b 的有向路径。

Warshall 更新式：

$$W_{ij}^{(k)} = W_{ij}^{(k-1)} \vee \left(W_{ik}^{(k-1)} \wedge W_{kj}^{(k-1)} \right). \tag{309}$$

含义是：允许使用前 k 个点作为中间点时，从 i 到 j 可达，要么原来就可达，要么通过第 k 个点可达。

9.3 等价关系与划分 / Equivalence Relations and Partitions

Theorem / 定理

关系 R 是等价关系，当且仅当 R 自反、对称、传递。

每个等价关系对应一个划分；每个划分也定义一个等价关系。

Equivalence class / 等价类

$$[a] = \{x \in A \mid xRa\}. \tag{310}$$

Proof / 证明

若 R 是等价关系，则每个元素属于自己的等价类；两个等价类若有交集，设 $z \in [a] \cap [b]$，由对称和传递可证 aRb，从而 [a]=[b]。因此等价类不交且覆盖 A，形成划分。反向：同一块中定义为等价，显然自反、对称、传递。

Use / 怎么用

模同余、字符串同长度、图中同连通分量都是等价关系。题目问“商集”或“等价类”时先找划分。

课堂展开：关系：等价类和偏序不要混 / Relations, Equivalence, and Posets

Example 1 / 等价关系

在整数集上定义 aRb 当且仅当 $a \equiv b \pmod 3$。证明它是等价关系。

Proof / 证明

自反：$a - a = 0$ 被 3 整除，所以 $a \equiv a \pmod 3$。

对称：若 $a \equiv b \pmod 3$，则 $3 \mid (a-b)$。于是 $3 \mid (b-a)$，所以 $b \equiv a \pmod 3$。

传递：若 $a \equiv b \pmod 3$ 且 $b \equiv c \pmod 3$，则 $3 \mid (a-b)$ 且 $3 \mid (b-c)$。相加得 $3 \mid (a-c)$，所以 $a \equiv c \pmod 3$。

等价类为：

$$[0] = \{\dots, -6, -3, 0, 3, 6, \dots\}, \tag{311}$$

$$[1] = \{\dots, -5, -2, 1, 4, 7, \dots\}, \tag{312}$$

$$[2] = \{\dots, -4, -1, 2, 5, 8, \dots\}. \tag{313}$$

Example 2 / 偏序

在 $\{1, 2, 3, 6\}$ 上用整除关系 $\mid$。这是偏序，因为自反、反对称、传递都成立。

Hasse 图的层次是：1 在底部，2 和 3 在中间，6 在顶部。2 与 3 不可比较。

How to use / 怎么用

等价关系强调“分组”；偏序强调“比较和层次”。看到自反、对称、传递是等价关系；看到自反、反对称、传递是偏序。

9.4 偏序 / Partial Orders

Definition / 定义

偏序关系是自反、反对称、传递的关系，记作 $\preceq$。

Examples / 例子

- $(\mathcal{P}(S), \subseteq)$.
- $(\mathbb{Z}^+, \mid)$.
- 任务依赖关系 / prerequisite relation.

Important elements / 重要元素

- maximal element / 极大元：没有更大的可比较元素。
- maximum element / 最大元：大于等于所有元素。
- minimal element / 极小元：没有更小的可比较元素。
- minimum element / 最小元：小于等于所有元素。
- upper bound / 上界：大于等于子集中所有元素。
- least upper bound / 最小上界：所有上界中的最小者。

Use / 怎么用

最大元一定是极大元；极大元不一定唯一，也不一定是最大元。画 Hasse 图时去掉自环、传递边，并把较大的元素画高。

深讲：偏序、格、上下确界 / Posets, Lattices, Bounds

Comparable / 可比较

在偏序集中，两个元素 a, b 可比较，指 $a \preceq b$ 或 $b \preceq a$。偏序不要求任意两元素可比较。

Least upper bound / 最小上界

子集 S 的上界 u 满足对所有 $s \in S$，$s \preceq u$。最小上界是所有上界中最小的那个，记作 $\mathrm{lub}(S)$ 或 join。

Greatest lower bound / 最大下界

下界 ℓ 满足对所有 $s \in S$，$\ell \preceq s$。最大下界记作 $\mathrm{glb}(S)$ 或 meet。

Lattice / 格

若偏序集中任意两个元素都有最小上界和最大下界，则称为格。

Example / 例子

在 $(\mathcal{P}(A), \subseteq)$ 中：

$$\mathrm{lub}(B, C) = B \cup C, \qquad \mathrm{glb}(B, C) = B \cap C. \tag{314}$$

在正整数整除偏序中：

$$\mathrm{lub}(a, b) = \mathrm{lcm}(a, b), \qquad \mathrm{glb}(a, b) = \gcd(a, b). \tag{315}$$

9.5 拓扑排序 / Topological Sorting

Theorem / 定理

有限偏序集总可以扩展成一个线性序。等价地，每个有限 DAG 都有拓扑排序。

Proof / 证明

有限 DAG 至少有一个入度为 0 的顶点；否则沿入边一直走会形成环。取该顶点作为第一个，删除它，对剩余 DAG 归纳。

Use / 怎么用

课程先修、任务调度、编译依赖都用拓扑排序。算法上反复取入度为 0 的点。

9.6 Dilworth 型结论 / Dilworth-Type Statements

Definitions / 定义

链 / chain 是任意两元素可比较的子集。反链 / antichain 是任意两个不同元素不可比较的子集。

Dilworth theorem / Dilworth 定理

有限偏序集中，最大反链大小等于把偏序集划分成链所需的最少链数。

Use / 怎么用

这类结论常用于调度、偏序分层、证明存在大型可比较或不可比较子集。若课程只讲基础版，通常需要会识别链和反链，不一定要求完整证明。

10. 图论 / Graph Theory

10.1 图的基本概念 / Basic Graph Concepts

Definition / 定义

无向图 $G = (V, E)$ 由顶点集 V 和边集 E 构成，边是顶点的无序对。简单图没有自环和重边。

常用概念：degree / 度、path / 路径、cycle / 圈、connected / 连通、component / 连通分量、subgraph / 子图、complete graph K_n、bipartite graph / 二分图。

模板：图论题流程 / Graph Problem Checklist

1. 先看是路径、圈、连通、平面、着色、匹配还是树。
2. 写出顶点数 v、边数 e、度数。
3. 用握手定理、Euler 公式、树边数公式或奇偶性。
4. 若要证明不存在，尝试反证加不变量或计数上界。

10.2 握手定理 / Handshake Lemma

Theorem / 定理

对任意无向图：

$$\sum_{v\in V}\deg(v)=2|E|. \tag{316}$$

因此奇度顶点个数为偶数。

Proof / 证明

每条边有两个端点，在度数总和中恰好被计数两次。由于总和是偶数，奇数项的个数必须为偶数。

Use / 怎么用

给度数序列判断是否可能、求边数、证明奇度点数量为偶数。

10.3 图同构 / Graph Isomorphism

Definition / 定义

图 G 与 H 同构，若存在双射 $f:V(G)\to V(H)$，使：

$$uv\in E(G)\iff f(u)f(v)\in E(H). \tag{317}$$

Use / 怎么用

证明同构：给出双射并验证邻接保持。证明不同构：找不变量不同，例如顶点数、边数、度数序列、连通分量、圈数量、割点数量。

补充：图的矩阵定理 / Matrix Theorems for Graphs

Adjacency matrix walk theorem / 邻接矩阵走路定理

设 A 是图 G 的邻接矩阵，则 A^k 的 (i,j) 元素等于从顶点 v_i 到 v_j 的长度为 k 的走路数量。

Proof / 证明

对 k 归纳。$k=1$ 时，邻接矩阵定义正好表示长度 1 的走路。假设 A^k 结论成立，则：

$$(A^{k+1})_{ij}=\sum_r (A^k)_{ir}A_{rj}. \tag{318}$$

这表示先从 v_i 走 k 步到某个 v_r，再沿一条边从 v_r 到 v_j。对所有中间点 r 求和，就是长度 $k+1$ 的走路数。

Complete graph edges / 完全图边数

简单完全图 K_n 的边数为：

$$\binom{n}{2}=\frac{n(n-1)}{2}. \tag{319}$$

Complete bipartite graph edges / 完全二分图边数

完全二分图 $K_{m,n}$ 的边数为：

$$mn. \tag{320}$$

补充：图同构判定常用不变量 / Graph Isomorphism Invariants

若两个图同构，则以下量必须相同：

- 顶点数。
- 边数。
- 度数序列。
- 连通分量个数。
- 是否二分图。
- 圈的数量和长度分布。
- 割点、桥的数量。
- 补图的对应性质。

Important / 重要点

这些条件通常只是必要条件，不是充分条件。两个图有相同度数序列也可能不同构。

Use / 怎么用

证明同构要给出保持邻接的双射；证明不同构通常找一个不变量不同。

10.4 连通性 / Connectivity

Definitions / 定义

G 连通表示任意两个顶点之间存在路径。点连通度 $\kappa(G)$ 是使图不连通或变成平凡图所需删除的最少顶点数。边连通度 $\lambda(G)$ 是使图不连通所需删除的最少边数。

Theorem / 基本不等式

对非平凡简单连通图：

$$\kappa(G) \leq \lambda(G) \leq \delta(G), \tag{321}$$

其中 $\delta(G)$ 是最小度。

Proof idea / 证明思路

删除某个最小度顶点的所有 关联边 可孤立它，所以 $\lambda(G) \leq \delta(G)$。点割通常不超过边割的难度，严格证明可由从最小边割构造点割得到。

Use / 怎么用

网络可靠性题常比较最小度、割边、割点。

10.5 欧拉路与欧拉回路 / Euler Paths and Circuits

Theorem / 欧拉回路判定

连通无向图有欧拉回路，当且仅当每个顶点度数为偶数。

Theorem / 欧拉路判定

连通无向图有欧拉路但无欧拉回路，当且仅当恰有两个奇度顶点。这两个奇度顶点是起点和终点。

Proof / 证明

必要性：欧拉回路每次进入一个顶点也必须离开它，关联边 成对使用，所以度数为偶。欧拉路的中间顶点仍成对，只有起点和终点各多一条边。充分性：从任意顶点沿未用边走，偶度保证除非回到起点不会卡住；若还有未用边，可把另一个闭合游走拼接进来。

Use / 怎么用

检查连通性和奇度顶点数量。题目问“一笔画”就是欧拉路问题。

课堂展开：图论：欧拉、哈密顿、平面图的区别 / Graph Problem Types

Euler vs Hamilton / 区别

欧拉问题关心边：是否能每条边恰好走一次。判定主要看奇度顶点。

哈密顿问题关心顶点：是否能每个顶点恰好访问一次。没有简单的完全判定公式，Dirac 和 Ore 只是充分条件。

Example / 欧拉路判断

若连通图的度数序列为：

$$3, 3, 2, 2, 2, 2, \tag{322}$$

奇度顶点有两个，因此有欧拉路但无欧拉回路。

若度数序列为：

$$4, 4, 2, 2, 2, \tag{323}$$

所有顶点度数为偶数，因此有欧拉回路。

Planar graph example / 平面图例题

证明 $K_{3,3}$ 非平面。

$K_{3,3}$ 有 $v = 6$ 个顶点，$e = 9$ 条边，且二分图没有三角形。若它平面，则由无三角形平面图边数上界：

$$e \leq 2v - 4 = 2 \cdot 6 - 4 = 8. \tag{324}$$

但 $e = 9 > 8$，矛盾。因此 $K_{3,3}$ 非平面。

How to use / 怎么用

图论题先判断是哪类问题，不要把欧拉路和哈密顿路混淆。欧拉看度数，哈密顿看顶点访问，平面图看 Euler 公式和边数上界。

深讲：欧拉定理的图论版本：为什么奇度点决定一笔画 / Euler Trails in Detail

Necessary condition / 必要性

在一条使用每条边恰好一次的 trail 中，除了起点和终点之外，每次到达某个中间顶点，都必须通过另一条未用边离开。因此中间顶点的 关联边 成对出现，度数为偶数。若起点等于终点，则所有顶点都是中间顶点，所以全偶。若起点不同于终点，则起点和终点各有一条“未配对”的边，所以恰有两个奇度点。

Sufficient condition idea / 充分性思路

若连通图所有顶点度数为偶数，从任意顶点出发不断走未用边。因为每个顶点未用边数保持偶配对，除非回到起点，否则不会卡住。得到一个闭合回路。如果还有未用边，由连通性可找到回路上某点连接未用边，从那里再走出一个闭合回路，并把两个回路拼接。重复直到所有边用完。

How to use / 怎么用

欧拉回路判定必须先忽略孤立点后检查连通性，再检查奇度点数。只有度数条件不够，图还要在有边部分连通。

10.6 哈密顿路与哈密顿圈 / Hamilton Paths and Cycles

Definitions / 定义

哈密顿路经过每个顶点恰好一次。哈密顿圈是闭合的哈密顿路。

Dirac theorem / Dirac 定理

若简单图 G 有 $n \geq 3$ 个顶点，且每个顶点度数至少 $n/2$，则 G 有哈密顿圈。

Ore theorem / Ore 定理

若对任意不相邻顶点 u, v 都有：

$$\deg(u) + \deg(v) \geq n, \tag{325}$$

则 G 有哈密顿圈。

Use / 怎么用

这些是充分条件，不是必要条件。不满足 Dirac/Ore 不能推出没有哈密顿圈。

10.7 最短路 / Shortest Paths

Dijkstra condition / Dijkstra 条件

Dijkstra 算法要求边权非负。

Invariant / 不变量

每次选出的未确定顶点 v 具有当前最小暂定距离时，该距离已经是从源点到 v 的最短距离。

Proof / 证明

若存在更短路径到 v，该路径第一次离开已确定集合时会经过某条非负边，因此到边界顶点的暂定距离不大于到 v 的更短距离，应先于 v 被选出，矛盾。

Use / 怎么用

有负边不要用 Dijkstra；可考虑 Bellman-Ford。无权图最短路用 BFS。

10.8 平面图 / Planar Graphs

Euler formula / 欧拉公式

连通平面图满足：

$$v - e + f = 2. \tag{326}$$

若有 c 个连通分量，则：

$$v - e + f = 1 + c. \tag{327}$$

Proof / 证明

对边数归纳。若图有环，删除环上一条边会减少一条边并合并两个面，所以 $v - e + f$ 不变。不断删到树。树有 $e = v - 1$、$f = 1$，所以 $v - e + f = 2$。

Corollaries / 推论

若简单连通平面图 $v \geq 3$，则：

$$e \leq 3v - 6. \tag{328}$$

若还没有三角形，则：

$$e \le 2v - 4. \tag{329}$$

Proof / 证明

每个面至少由 3 条边围成，每条边最多被两个面计数，所以 $3f \le 2e$。代入欧拉公式得 $e \le 3v - 6$。无三角形时每个面至少 4 条边，故 $4f \le 2e$，得 $e \le 2v - 4$。

Use / 怎么用

证明非平面常用反证：假设平面，代入边数上界得到矛盾。K_5 因 $e = 10 > 3 \cdot 5 - 6 = 9$ 非平面；$K_{3,3}$ 因无三角形且 $e = 9 > 2 \cdot 6 - 4 = 8$ 非平面。

补充：Kuratowski、五色定理、四色定理 / Planar Graph Theorems

Kuratowski theorem / Kuratowski 定理

图是平面图，当且仅当它不包含与 K_5 或 $K_{3,3}$ 的细分同胚的子图。

Meaning / 含义

细分一条边就是在边中间插入若干度数为 2 的顶点。Kuratowski 定理说明，所有非平面性的根源都来自 K_5 或 $K_{3,3}$。

Five color theorem / 五色定理

每个平面图都可以用至多 5 种颜色进行正常顶点染色。

Proof idea / 证明思路

由平面图边数上界可知，每个平面图都有度数不超过 5 的顶点。删除该顶点，对剩余图归纳染 5 色。若邻居用了少于 5 种颜色，直接给删去的顶点用剩余颜色；若五个邻居用满五色，需要用 Kempe chain 交换颜色腾出一种颜色。

Four color theorem / 四色定理

每个平面图都可以用至多 4 种颜色染色。该定理证明很深，通常课程只要求知道结论和用途，不要求掌握证明。

深讲：平面图：面计数为什么是 $2e$ / Planar Graph Counting

Face-degree sum / 面度数和

在平面嵌入中，把每个面的边界长度加起来。每条边通常出现在两个面的边界中；桥会在同一个面的边界中走两次。总之每条边贡献 2，所以：

$$\sum_{\text{faces } f} \deg(f) = 2e. \tag{330}$$

Why $e \le 3v - 6$ / 为什么有边数上界

若简单平面图且 $v \ge 3$，每个面边界长度至少为 3。因此：

$$3f \le 2e. \tag{331}$$

由 Euler 公式 $v - e + f = 2$，得 $f = 2 - v + e$。代入：

$$3(2 - v + e) \le 2e. \tag{332}$$

整理：

$$e \le 3v - 6. \tag{333}$$

Triangle-free case / 无三角形情况

若没有三角形，每个面至少 4 条边，所以：

$$4f \le 2e. \tag{334}$$

代入 Euler 公式得到：

$$e \le 2v - 4. \tag{335}$$

How to use / 怎么用

证明非平面时，先看有没有三角形。若无三角形，用更强的 $e \le 2v-4$。

10.9 图着色 / Graph Coloring

Definition / 定义

图着色是给顶点染色，使相邻顶点颜色不同。色数 $\chi(G)$ 是所需最少颜色数。

Basic bounds / 基本界

$$\omega(G) \le \chi(G) \le \Delta(G) + 1, \tag{336}$$

其中 $\omega(G)$ 是最大团大小，$\Delta(G)$ 是最大度。

Proof / 证明

团中任意两点相邻，必须用不同颜色，所以 $\omega(G) \le \chi(G)$。贪心染色时，每个顶点最多有 $\Delta(G)$ 个已染邻居，所以总能从 $\Delta(G)+1$ 种颜色中找到一种可用。

Theorem / 二分图判定

图是二分图当且仅当它没有奇圈。

Proof / 证明

若图二分，则任何圈在两侧交替，长度必为偶数。反过来，若没有奇圈，从每个连通分量任选根，按到根的距离奇偶分两组；若同组有边，则形成奇圈，矛盾。

Use / 怎么用

课程排课、寄存器分配、冲突图都可建模为着色。二分图常用 BFS 两色染色判定。

补充：色多项式 / Chromatic Polynomial

Definition / 定义

图 G 的色多项式 $P_G(k)$ 表示用 k 种颜色对 G 做正常顶点染色的方案数。

Examples / 例子

空边图 E_n：

$$P_{E_n}(k) = k^n. \tag{337}$$

完全图 K_n：

$$P_{K_n}(k) = k(k-1)(k-2)\cdots(k-n+1). \tag{338}$$

树 T 有 n 个顶点：

$$P_T(k) = k(k-1)^{n-1}. \tag{339}$$

Deletion-contraction / 删除-收缩递推

对非环边 e：

$$P_G(k) = P_{G-e}(k) - P_{G/e}(k). \tag{340}$$

Use / 怎么用

色多项式题常用删除-收缩递推，或直接套特殊图公式。

深讲：二分图、奇圈和 BFS 染色 / Bipartite Graphs in Detail

Theorem / 定理

图 G 是二分图，当且仅当 G 不含奇圈。

Proof: bipartite implies no odd cycle / 二分图推出无奇圈

若图可分成左右两侧 L 和 R，每条边都跨越两侧。沿任何圈走，顶点所在侧必须左右交替。要回到起点，走的边数必须为偶数。因此不存在奇圈。

Proof: no odd cycle implies bipartite / 无奇圈推出二分图

假设图连通。选一个根 s，令：

$$L = \{v : d(s,v) \text{ is even}\}, \qquad R = \{v : d(s,v) \text{ is odd}\}. \tag{341}$$

若存在一条边连接同一侧两个顶点 u, v，则 $d(s,u)$ 和 $d(s,v)$ 奇偶相同。把从 s 到 u 的最短路、边 uv、从 v 回到 s 的最短路合起来，可得到一个奇闭合走路，其中包含奇圈，矛盾。因此每条边都跨越 L, R，图是二分图。非连通图对每个分量分别做。

Algorithm / 算法

BFS 两色染色：从未染色顶点开始染红，邻居染蓝，邻居的邻居染红。若某条边两端颜色相同，则不是二分图。

深讲：图着色：上下界和贪心算法 / Coloring Bounds

Clique lower bound / 团给下界

若图中有 k 个顶点两两相邻，则这 k 个顶点必须使用 k 种不同颜色，所以：

$$\chi(G) \geq k. \tag{342}$$

最大团大小给出：

$$\chi(G) \geq \omega(G). \tag{343}$$

Greedy upper bound / 贪心给上界

按任意顺序给顶点染色。染某个顶点时，它最多有 Δ 个邻居，因此最多禁用 Δ 种颜色。如果准备 $\Delta + 1$ 种颜色，总有一种颜色可用。所以：

$$\chi(G) \leq \Delta(G) + 1. \tag{344}$$

How to use / 怎么用

求色数通常要上下界夹逼。先找团说明至少要多少色，再构造染色说明至多要多少色。如果上下界相等，就得到色数。

10.10 匹配、Hall 定理与稳定婚姻 / Matching, Hall, and Stable Marriage

Definition / 定义

匹配是没有共享端点的边集。二分图 $G = (L \cup R, E)$ 中，覆盖 L 的匹配叫 覆盖左部的匹配。

Hall's theorem / Hall 定理

二分图存在覆盖 L 的匹配，当且仅当对所有 $S \subseteq L$：

$$|N(S)| \geq |S|. \tag{345}$$

Proof idea / 证明思路

必要性显然：S 中每个点要匹配到不同邻居。充分性可用增广路或归纳证明。

Use / 怎么用

分配题、婚配题、任务指派题常用 Hall 条件。证明不存在完美匹配时找一个 S 使 $|N(S)| < |S|$。

Stable marriage / 稳定婚姻

Gale-Shapley deferred acceptance algorithm 总会终止并返回稳定匹配。

Proof idea / 证明思路

每轮有人向尚未拒绝过的对象求婚，总求婚次数有限。稳定性来自：若存在阻塞对，则其中一方曾向另一方求婚，另一方只会保留更喜欢的对象，因此不可能双方都更喜欢彼此。

补充：稳定婚姻的更强结论 / Stronger Stable Marriage Results

Stable matching exists / 稳定匹配存在

Gale-Shapley 延迟接受算法一定终止，并输出稳定匹配。

Proposer-optimal theorem / 提议方最优定理

若由左侧参与者提议，则 Gale-Shapley 输出的稳定匹配对所有左侧参与者都是其在所有稳定匹配中能得到的最好对象；同时对右侧参与者是最差稳定对象。

Proof idea / 证明思路

一个提议者只有在被拒绝后才会向下一个选择提议。若某提议者被一个对象拒绝，说明该对象当前持有或未来会持有一个她更喜欢且可稳定匹配的提议者。因此被拒绝的对象不可能是该提议者在任何稳定匹配中的可行最好对象。归纳所有拒绝可得提议方最优。

Use / 怎么用

稳定婚姻题如果问“算法输出是否唯一”，答案通常是否；但若固定提议方，Gale-Shapley 输出的是提议方最优稳定匹配。

10.11 通信网络指标 / Communication Network Metrics

Useful quantities / 常用指标

- distance $d(u, v)$: shortest path length.
- diameter $\max_{u,v} d(u, v)$.
- average degree $2|E|/|V|$.
- clustering coefficient: local triangle density.
- cut / min-cut: reliability bottleneck.

Use / 怎么用

网络题通常把“效率”翻译成距离或直径，把“可靠性”翻译成连通度或割，把“局部紧密程度”翻译成三角形或聚类系数。

补充：MIT 通信网络常用公式 / MIT Communication Network Formulas

Complete binary tree network / 完全二叉树网络

若有 $N = 2^n$ 个叶子，则树高为 $n = \log_2 N$。叶子之间最远距离为：

$$2 \log_2 N. \tag{346}$$

内部交换节点数为：

$$N-1. \tag{347}$$

2-D array / 二维网格网络

若有 $N=m^2$ 个节点排成 $m\times m$ 网格，则直径为：

$$2(m-1)=\Theta(\sqrt{N}). \tag{348}$$

Butterfly network / 蝶形网络

若 $N=2^n$，蝶形网络通常有 $n+1$ 层，每层 N 个节点，总节点量级为：

$$\Theta(N\log N). \tag{349}$$

从输入到输出的路径长度为：

$$\log_2 N. \tag{350}$$

Use / 怎么用

MIT 通信网络题常比较 diameter / 直径、switch count / 交换节点数、latency / 延迟、congestion / 拥塞。不要只看节点数，路由长度和拥塞也很关键。

11. 树 / Trees

11.1 树的等价刻画 / Equivalent Characterizations

Theorem / 定理

对无向简单图 G，以下条件等价：

1. G 是树。
2. G 连通且无圈。
3. G 连通且 $|E|=|V|-1$。
4. G 无圈且 $|E|=|V|-1$。
5. 任意两个顶点之间有唯一简单路径。
6. G 极小连通：删任意边都不连通。
7. G 极大无圈：加任意非边都会产生圈。

Proof / 证明思路

核心是“无圈图每个连通分量是树，边数为顶点数减一”。连通无圈保证路径存在且唯一；若两点有两条不同简单路径，则合起来形成圈。边数公式可对顶点数归纳：树中一定有叶子，删叶子后仍是树。

Use / 怎么用

树题常在这些刻画间切换。要证明某图是树，最常用“连通且边数 $n-1$”或“连通且无圈”。

课堂展开：树：为什么 $e=v-1$ 这么常用 / Trees in Detail

Detailed proof / 详细证明

证明每棵有 v 个顶点的树有 $v-1$ 条边。

对 v 做归纳。若 $v=1$，树只有一个顶点没有边，所以 $e=0=v-1$。

假设所有 $v=k$ 的树都有 $k-1$ 条边。考虑 $v=k+1$ 的树。树至少有一个叶子，删除一个叶子及其 关联边 后，剩余图仍连通且无圈，所以仍是树，有 k 个顶点。由归纳假设，剩余树有 $k-1$ 条边。加回叶子和那条边，总边数为：

$$(k-1)+1=k=(k+1)-1. \tag{351}$$

所以结论成立。

Use / 怎么用

只要题目给出连通图且边数为 $v-1$，通常可以立刻想到树。只要题目给出无圈图且边数为 $v-1$，也通常可以推出树。

深讲：树的七个等价条件怎么互相推 / Tree Equivalences

Core cycle-path principle / 核心原则

在无向图中，若两个顶点之间存在两条不同简单路径，则这两条路径合起来包含一个圈。反过来，若图中有圈，则圈上任意两个顶点之间有两条不同路径。

Tree implies unique path / 树推出唯一路径

树连通，所以任意两点之间至少有一条路径。若有两条不同简单路径，则会形成圈，矛盾。因此路径唯一。

Unique path implies tree / 唯一路径推出树

若任意两点有唯一路径，则图连通。若图中有圈，则圈上两点之间有两条不同路径，矛盾。所以无圈。连通且无圈，因此是树。

Minimal connected / 极小连通

若树中删除任意边 uv，原来 u 到 v 的唯一路径就是这条边。删除后 u,v 不再连通，所以图不连通。

Maximal acyclic / 极大无圈

若在树中加入一条非边 uv，原树中已有从 u 到 v 的唯一路径。加入 uv 后，这条路径加新边形成一个圈。

11.2 叶子与度数公式 / Leaves and Degree Formula

Theorem / 叶子定理

每个至少有两个顶点的树至少有两个叶子。

Proof / 证明

取树中最长简单路径。若端点度数大于 1，则可从端点延伸到路径外或路径内形成更长路径或圈，矛盾。因此两个端点都是叶子。

Degree formula / 度数公式

若树有 n 个顶点，则：

$$\sum_{v\in V}\deg(v)=2(n-1). \tag{352}$$

若 n_i 表示度数为 i 的顶点个数，则：

$$\sum_i n_i=n, \qquad \sum_i i n_i=2(n-1). \tag{353}$$

Use / 怎么用

给定树的度数分布求叶子数时，用这两个方程。

11.3 根树与 m 叉树 / Rooted and m-ary Trees

Definitions / 定义

根树有一个特殊顶点 root。父、子、祖先、后代、层数、高度都相对根定义。满 m 叉树中每个内部顶点都有 m 个孩子。

Theorem / 满 m 叉树公式

若满 m 叉树有 i 个内部顶点、ℓ 个叶子、总顶点 n，则：

$$n = mi + 1, \qquad \ell = (m-1)i + 1, \qquad n = \frac{m\ell - 1}{m-1}. \tag{355}$$

Proof / 证明

每个内部顶点产生 m 条到孩子的边，所以边数 $e = mi$。树总有 $e = n - 1$，因此 $n = mi + 1$。又 $n = i + \ell$，代入得叶子公式。

Use / 怎么用

编码树、决策树、堆结构、表达式树中常用这些公式。

补充：二叉树高度与叶子界 / Binary and m-ary Tree Bounds

Height bound / 高度界

高度为 h 的 m 叉树最多有：

$$m^h \tag{356}$$

个叶子。

Proof / 证明

第 0 层最多 1 个顶点，第 1 层最多 m 个，第 h 层最多 m^h 个。叶子最多全在最深层，因此最多 m^h 个。

Leaf-height lower bound / 叶子数推出高度下界

若 m 叉树有 ℓ 个叶子，高度为 h，则：

$$\ell \le m^h, \tag{357}$$

所以：

$$h \ge \lceil \log_m \ell \rceil. \tag{358}$$

Full binary tree / 满二叉树

满二叉树中每个内部顶点恰有 2 个孩子。若内部顶点数为 i，叶子数为 ℓ，则：

$$\ell = i + 1. \tag{359}$$

这是满 m 叉树公式 $\ell = (m-1)i + 1$ 在 $m = 2$ 时的特例。

11.4 遍历 / Traversal

Orders / 顺序

- Preorder / 前序: 根、左子树、右子树.
- Inorder / 中序: left, root, right.
- Postorder / 后序: left, right, root.
- Level order / 层序: breadth-first.

Use / 怎么用

表达式树：中序接近普通表达式，后序对应逆波兰表达式，前序对应前缀表达式。

11.5 生成树与最小生成树 / Spanning Trees and MST

Theorem / 生成树存在

每个连通图都有生成树。

Proof / 证明

若连通图有圈，删除圈上一条边仍连通。重复删除直到无圈，得到连通无圈的生成子图，即生成树。

Cut property / 割性质

在加权连通图中，对任意割，跨越该割的最轻边属于某棵最小生成树。

Proof / 证明

取一棵 MST。若它不含该最轻边 e，加入 e 会形成圈，圈中必有另一条跨同一割的边 f。用 e 替换 f 不会增加权重，仍得到 MST。

Use / 怎么用

Kruskal 和 Prim 的正确性都依赖割性质。Kruskal 每次选不成圈的最轻边；Prim 每次从已选顶点集合向外选最轻边。

课堂展开：最小生成树：Kruskal 为什么正确 / MST Greedy Correctness

Kruskal algorithm / Kruskal 算法

把所有边按权重从小到大排序，依次选择不会形成圈的边，直到选出 $v-1$ 条边。

Why it works / 为什么正确

Kruskal 每次选择的边都跨越某个割，并且是该割上的最轻可用边。根据割性质，这条边可以属于某棵 MST。不断选择这样的安全边，最终得到一棵 MST。

Example / 例题步骤

做 MST 计算题时写三列：候选边、是否形成圈、是否选择。每次遇到形成圈的边就跳过。最后检查是否选了 $v-1$ 条边；若少于 $v-1$，原图不连通。

11.6 Huffman 编码 / Huffman Coding

Theorem / Huffman 贪心选择性质

最优前缀编码树中，频率最低的两个符号可以作为最深层的兄弟叶子。

Proof idea / 证明思路

在任一最优树中，最深兄弟叶子若不是最低频率符号，可与低频符号交换，不会增加加权路径长度。于是可把两个最低频率符号合并为一个伪符号，递归求解。

Use / 怎么用

构造 Huffman 树时反复合并两个最小权重。证明最优性时写 贪心选择性质 + 最优子结构。

深讲：Huffman 编码的交换论证 / Huffman Exchange Argument

Goal / 目标

Huffman 算法每次合并两个最低频率符号。要证明贪心正确，需要证明存在某棵最优树也把这两个最低频率符号放在最深处且互为兄弟。

Exchange argument / 交换论证

在任意最优前缀码树中，取最深层的一对兄弟叶子 x, y。它们的深度最大。若最低频率符号 a, b 不在这两个位置，把 a, b 与 x, y 交换。因为 a, b 的频率不高于 x, y，把低频符号放到更深处不会增加加权路径长度：

$$\sum_i f_i d_i. \tag{360}$$

所以存在一棵最优树让两个最低频率符号成为最深兄弟。把它们合并成一个权重为 $f_a + f_b$ 的伪符号后，问题规模减少 1，递归成立。

How to use / 怎么用

证明贪心算法时常用 exchange argument：说明任意最优解都能改造成包含贪心选择的最优解。

12. 布尔代数 / Boolean Algebra

12.1 布尔代数公理与对偶 / Boolean Algebra and Duality

Operations / 运算

布尔代数通常有 $+$、$\cdot$、$\overline{x}$，分别对应 OR、AND、NOT。

Laws / 公式

$$\begin{aligned} x + 0 &= x, & x \cdot 1 &= x, \\ x + 1 &= 1, & x \cdot 0 &= 0, \\ x + x &= x, & x \cdot x &= x, \\ x + \overline{x} &= 1, & x \cdot \overline{x} &= 0, \\ \overline{x + y} &= \overline{x} \cdot \overline{y}, \\ \overline{x \cdot y} &= \overline{x} + \overline{y}. \end{aligned} \tag{361}$$

Duality / 对偶原理

若一个布尔恒等式成立，把 $+$ 与 $\cdot$ 互换、0 与 1 互换后得到的对偶式也成立。

Use / 怎么用

化简逻辑电路和布尔表达式。对偶原理可减少记忆量。

12.2 布尔函数表示 / Representing Boolean Functions

Theorem / DNF and CNF

每个布尔函数都可表示为析取范式 DNF，也可表示为合取范式 CNF。

Proof / 证明

DNF：对真值表中每一行输出 1 的赋值，写一个 minterm，使它只在该行取 1；所有 minterm 取 OR。CNF 类似，对输出 0 的每行写 maxterm。

Use / 怎么用

真值表转表达式时，输出 1 行少用 DNF；输出 0 行少用 CNF。

课堂展开：布尔函数：从真值表写 DNF / Boolean Functions from Truth Tables

Example / XOR 的 DNF

异或 $p \oplus q$ 在两种情况下为 1：

$$(p, q) = (1, 0) \quad \text{or} \quad (p, q) = (0, 1). \tag{362}$$

对应 minterms：

$$p \wedge \neg q, \qquad \neg p \wedge q. \tag{363}$$

所以：

$$p \oplus q \equiv (p \wedge \neg q) \vee (\neg p \wedge q). \tag{364}$$

How to use / 怎么用

从真值表写 DNF：只看输出为 1 的行；每行写一个“完全匹配该行”的 AND 项；最后 OR 起来。

12.3 功能完备性 / Functional Completeness

Theorem / NAND 与 NOR 完备

NAND 单独功能完备，NOR 单独功能完备。

Proof / NAND

$$\neg x = x \uparrow x, \qquad x \wedge y = (x \uparrow y) \uparrow (x \uparrow y), \qquad x \vee y = (x \uparrow x) \uparrow (y \uparrow y). \tag{365}$$

由于 NOT、AND、OR 可由 NAND 表示，而它们能表示任意布尔函数，所以 NAND 完备。NOR 类似。

Use / 怎么用

电路设计中若题目限制只能使用 NAND 或 NOR，就先把基本门转换出来。

12.4 Karnaugh 图 / Karnaugh Maps

Idea / 思路

卡诺图 / K-map 把相邻只差一个变量的 minterm 放在相邻格。把 1 按 $1, 2, 4, 8, \ldots$ 个相邻格分组，消去变化变量，得到简化 DNF。

Use / 怎么用

适合 2 到 4 个变量的手算化简。注意边界相邻和角落相邻。

深讲：布尔代数：最小项、最大项、卡诺图 / Minterms, Maxterms, K-Maps

Minterm / 最小项

对 n 个变量，每个 minterm 是所有变量或其否定的 AND，且只在一行真值表中取 1。例如变量 x, y, z 中，行 $(1, 0, 1)$ 对应：

$$x \wedge \neg y \wedge z. \tag{366}$$

Maxterm / 最大项

maxterm 是变量或其否定的 OR，且只在一行真值表中取 0。若某行是 $(1, 0, 1)$，对应 maxterm 为：

$$\neg x \vee y \vee \neg z. \tag{367}$$

因为在该行中每一项都为 0。

DNF and CNF / 析取范式与合取范式

DNF 把所有输出为 1 的行对应 minterm OR 起来。CNF 把所有输出为 0 的行对应 maxterm AND 起来。

卡诺图分组 / K-map grouping

卡诺图的相邻格只差一个变量。把相邻的 1 合并时，变化的变量被消去，不变的变量保留。分组大小必须是 $1, 2, 4, 8, \ldots$。

13. 计算模型 / Modeling Computation

13.1 语言与文法 / Languages and Grammars

Definitions / 定义

字母表 Σ 是符号集合。字符串是有限符号序列。语言是 Σ^* 的子集。

文法由变量、终结符、产生式和开始符号组成。若字符串可由开始符号推导出，则属于该文法生成的语言。

Use / 怎么用

证明字符串属于语言：给推导过程。证明不属于语言：说明任何推导都无法满足某个结构不变量。

补充：文法类型 / Grammar Types

Phrase-structure grammar / 短语结构文法

产生式形式最一般，左边可以是含变量的字符串。

Context-sensitive grammar / 上下文有关文法

产生式不能缩短字符串，常见形式为：

$$\alpha A\beta \to \alpha\gamma\beta. \tag{368}$$

Context-free grammar / 上下文无关文法

每条产生式左边只有一个变量：

$$A \to \gamma. \tag{369}$$

Regular grammar / 正则文法

产生式形如：

$$A \to aB, \qquad A \to a, \qquad A \to \lambda. \tag{370}$$

Hierarchy / 层级

$$\text{regular} \subseteq \text{context-free} \subseteq \text{context-sensitive} \subseteq \text{phrase-structure}. \tag{371}$$

Use / 怎么用

正则语言对应有限自动机；上下文无关语言常用于程序语言语法和括号匹配。

13.2 有限状态机 / Finite-State Machines

Definition / 定义

确定有限自动机 DFA 为五元组：

$$(Q, \Sigma, \delta, q_0, F), \tag{372}$$

其中 Q 是有限状态集，Σ 是字母表，$\delta : Q \times \Sigma \to Q$ 是转移函数，q_0 是初态，F 是接受态集合。

Use / 怎么用

构造自动机时，状态通常表示“到目前为止需要记住的信息”，例如余数、奇偶性、是否见过某模式。

补充：有输出有限状态机 / Finite-State Machines with Output

Mealy machine / Mealy 机

Mealy 机的输出依赖当前状态和输入：

$$M = (S, I, O, f, g, s_0), \tag{373}$$

其中 $f : S \times I \to S$ 是状态转移函数，$g : S \times I \to O$ 是输出函数。

Moore machine / Moore 机

Moore 机的输出只依赖当前状态：

$$M = (S, I, O, f, g, s_0), \tag{374}$$

其中 $g : S \to O$。

区别 / Difference

Mealy 输出写在边上，Moore 输出写在状态上。Mealy 通常响应更快，Moore 通常更容易分析。

Use / 怎么用

Rosen 的 computation modeling 中会区分“有输出状态机”和“无输出状态机”。识别语言通常用无输出自动机；建模电路、售货机、协议时常用有输出状态机。

课堂展开：自动机：状态到底表示什么 / Finite Automata Intuition

Example / 例题

构造 DFA 接受所有二进制串，使其表示的二进制数能被 3 整除。

State idea / 状态含义

读入前缀后，只需要记住当前数模 3 的余数。因此状态为：

$$q_0 : \text{remainder } 0, \quad q_1 : \text{remainder } 1, \quad q_2 : \text{remainder } 2. \tag{375}$$

读入一个新 bit $b \in \{0, 1\}$ 时，原数 x 变成 $2x + b$，所以余数更新为：

$$r' \equiv 2r + b \pmod 3. \tag{376}$$

接受态是 q_0。

Transition table / 转移表

	0	1
q_0	q_0	q_1
q_1	q_2	q_0
q_2	q_1	q_2

(377)

How to use / 怎么用

自动机题的关键是“状态存什么信息”。能被 m 整除的问题通常存余数；奇偶性问题存 parity；是否出现某个模式的问题存已经匹配到的最长前缀。

13.3 正则表达式与自动机 / Regular Expressions and Automata

Kleene theorem / Kleene 定理

一个语言可被有限自动机识别，当且仅当它可由正则表达式描述。

$$L \text{ is regular} \iff L \text{ is recognized by a finite automaton} \iff L \text{ is described by a regular expression.} \tag{378}$$

Proof idea / 证明思路

从正则表达式到 NFA：对 union、concatenation、star 做结构归纳构造。自动机到正则表达式：逐步消去状态，把路径标签合并成正则表达式。

Use / 怎么用

正则语言题可在 regex、DFA、NFA 三种表示间切换，选最方便的一种。

深讲：正则语言和 Pumping Lemma / Regular Languages and Pumping Lemma

Pumping lemma / 泵引理

若语言 L 是正则语言，则存在泵长度 p，使得任意 $s \in L$ 且 $|s| \geq p$，都可写成：

$$s = xyz, \tag{379}$$

满足：

$$|xy| \leq p, \qquad |y| > 0, \qquad xy^i z \in L \quad \text{for all } i \geq 0. \tag{380}$$

Why it holds / 为什么成立

正则语言可由 DFA 识别。若 DFA 有 p 个状态，读入长度至少 p 的字符串前 p 个符号时，根据鸽巢原理，某个状态必重复。重复状态之间读到的子串就是 y。因为从同一状态绕这个环任意多次仍能继续到接受态，所以 y 可以被 泵送。

How to use / 怎么用

证明语言非正则时使用反证：假设正则，取 泵长度 p，选一个特别的字符串 $s \in L$，证明无论如何分解 $s = xyz$，都能找到某个 i 使 $xy^i z \notin L$。

Example / 例题

证明：

$$L = \{0^n 1^n : n \geq 0\} \tag{381}$$

不是正则语言。

假设 L 正则，取 泵长度 p。令：

$$s = 0^p 1^p. \tag{382}$$

因为 $|xy| \leq p$，所以 x 和 y 都只包含前面的 0，且 $|y| > 0$。把 y 泵降，即取重复次数 0，即取 $i = 0$，得到的字符串 xz 中 0 的数量少了，但 1 的数量仍为 p，所以不再属于 L。矛盾。因此 L 非正则。

13.4 图灵机与不可判定性 / Turing Machines and Undecidability

Core theorem / 停机问题不可判定

不存在算法能对所有程序 P 和输入 x 判断 $P(x)$ 是否停机。

Proof / 证明

反证。假设存在判定器 $H(P, x)$，若 $P(x)$ 停机则返回 true，否则 false。构造程序 $D(P)$：若 $H(P, P)$ 为 true，则进入无限循环；若为 false，则停机。现在问 $D(D)$。若 $H(D, D)$ true，则 $D(D)$ 不停机；若 false，则 $D(D)$ 停机。矛盾。

Use / 怎么用

不可判定性证明常用自指或归约。若能用假想算法解决停机问题，则该假想算法不存在。

课堂展开：停机问题：反证法的自指结构 / Halting Problem Explained

Goal / 目标

证明不存在通用程序 $H(P,x)$，能判断任意程序 P 在输入 x 上是否停机。

Contradiction setup / 反证设置

假设 H 存在。定义新程序 $D(P)$：

- 若 $H(P,P)$ 判断 $P(P)$ 会停机，则 $D(P)$ 故意进入死循环。
- 若 $H(P,P)$ 判断 $P(P)$ 不会停机，则 $D(P)$ 立刻停机。

现在考虑 $D(D)$。

若 $H(D,D)$ 说 $D(D)$ 会停机，则按照 D 的定义，$D(D)$ 会死循环，矛盾。

若 $H(D,D)$ 说 $D(D)$ 不会停机，则按照 D 的定义，$D(D)$ 会立刻停机，矛盾。

两种情况都矛盾，所以 H 不存在。

How to use / 怎么用

不可判定性证明通常都是：假设有一个万能判定器，然后构造一个“故意反着做”的对象，让它在自己身上运行。

14. MIT 6.042J / 6.1200J 常见补充专题

14.1 状态机 / State Machines

Definition / 定义

状态机由状态集合、初始状态、转移规则组成。可达状态是从初始状态经过有限步转移能到达的状态。

Invariant theorem / 不变量定理

若 I 对所有初始状态成立，并且每个合法转移都保持 I，则 I 对所有可达状态成立。

Use / 怎么用

MIT 的很多证明把算法、游戏、协议都看成状态机。证明“坏状态不可达”时找一个不变量排除坏状态。

14.2 随机游走 / Random Walks

Simple random walk / 简单随机游走

在整数线上，每步以概率 $1/2$ 向左或向右。若 X_t 是第 t 步位置，则：

$$\mathbb{E}[X_t] = 0, \qquad \mathrm{Var}(X_t) = t. \tag{383}$$

Proof / 证明

令每步增量 Y_i 取 1 或 -1，且独立同分布。则 $X_t = \sum_{i=1}^{t} Y_i$，每个 $\mathbb{E}[Y_i] = 0$，$\mathrm{Var}(Y_i) = 1$。由期望线性性和独立方差可加性得到结论。

Use / 怎么用

随机过程题通常先拆成指示变量或独立增量，再用期望和方差公式。

14.3 游戏与必败态 / Games and Losing Positions

Nim xor rule / Nim 异或规则

普通取石子 Nim 游戏中，局面为 $a_1, \dots, a_k$。若：

$$a_1 \oplus a_2 \oplus \cdots \oplus a_k = 0, \tag{384}$$

则该局面是必败态；否则是必胜态。

Proof idea / 证明思路

若 xor 为 0，任意移动都会改变某一堆，使 xor 非零。若 xor 非零，取最高位为 1 的位，选择一堆在该位为 1 的石子，把它减少到使总 xor 变 0。因此非零局面可走到零局面，零局面只能走到非零局面。

Use / 怎么用

遇到 impartial combinatorial game，先找 P-position / N-position。Nim 直接算 xor。

15. 公式索引 / Formula Index

15.1 Logic / 逻辑

- $p \to q \equiv \neg p \lor q$
- $p \to q \equiv \neg q \to \neg p$
- $p \leftrightarrow q \equiv (p \to q) \land (q \to p)$
- $\neg \forall x P(x) \equiv \exists x \neg P(x)$
- $\neg \exists x P(x) \equiv \forall x \neg P(x)$

15.2 Sets / 集合

- $A = B \iff A \subseteq B \land B \subseteq A$
- $(A \cap B)^c = A^c \cup B^c$
- $(A \cup B)^c = A^c \cap B^c$
- $|\mathcal{P}(A)| = 2^{|A|}$
- $|A \cup B| = |A| + |B| - |A \cap B|$

15.3 Sums / 求和

- $\sum_{i=1}^{n} i = n(n+1)/2$
- $\sum_{i=1}^{n} i^2 = n(n+1)(2n+1)/6$
- $\sum_{i=1}^{n} i^3 = (n(n+1)/2)^2$
- $\sum_{i=0}^{n} r^i = (r^{n+1}-1)/(r-1)$
- $H_n = \Theta(\log n)$

15.4 Asymptotics / 渐近

- $f = O(g)$: eventually $f \le Cg$
- $f = \Omega(g)$: eventually $f \ge Cg$
- $f = \Theta(g)$: both O and Ω
- $1 \prec \log n \prec n^c \prec a^n \prec n!$

15.5 Number Theory / 数论

- $a \equiv b \pmod{m} \iff m \mid (a-b)$
- $\gcd(a,b) = \gcd(b, a \bmod b)$
- $\gcd(a,b)\,\mathrm{lcm}(a,b) = |ab|$
- Bezout: $\gcd(a,b) = sa + tb$
- a 在模 m 下有逆元，当且仅当 $\gcd(a,m) = 1$
- CRT: $x \equiv \sum_i a_i M_i y_i \pmod{M}$
- Fermat: $a^{p-1} \equiv 1 \pmod{p}$

- Euler: $a^{\varphi(n)} \equiv 1 \pmod{n}$

15.6 Counting / 计数

- $P(n,r) = n!/(n-r)!$
- $\binom{n}{r} = n!/(r!(n-r)!)$
- $\sum_{r=0}^{n} \binom{n}{r} = 2^n$
- $(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^{n-k} y^k$
- Stars and bars: $\binom{n+k-1}{k-1}$
- Derangements: $D_n = n! \sum_{k=0}^{n} (-1)^k / k!$

15.7 Probability / 概率

- $\Pr(A^c) = 1 - \Pr(A)$
- $\Pr(A \cup B) = \Pr(A) + \Pr(B) - \Pr(A \cap B)$
- $\Pr(A \mid B) = \Pr(A \cap B) / \Pr(B)$
- Bayes: $\Pr(B_j \mid A) = \Pr(A \mid B_j) \Pr(B_j) / \sum_i \Pr(A \mid B_i) \Pr(B_i)$
- $\mathbb{E}[X] = \sum_x x \Pr(X = x)$
- $\mathrm{Var}(X) = \mathbb{E}[X^2] - \mathbb{E}[X]^2$

15.8 Recurrences / 递推

- $a_n = ca_{n-1} + d \Rightarrow a_n = c^n a_0 + d(c^n - 1)/(c-1)$
- $a_n = c_1 a_{n-1} + c_2 a_{n-2}$ uses characteristic equation $r^2 - c_1 r - c_2 = 0$
- Master theorem: compare $f(n)$ with $n^{\log_b a}$
- $A(x) = \sum_{n \geq 0} a_n x^n$

15.9 Relations / 关系

- Equivalence relation = reflexive + symmetric + transitive.
- Partial order = reflexive + antisymmetric + transitive.
- Transitive closure update: $W_{ij}^{(k)} = W_{ij}^{(k-1)} \vee (W_{ik}^{(k-1)} \wedge W_{kj}^{(k-1)})$

15.10 Graphs and Trees / 图与树

- Handshake: $\sum_v \deg(v) = 2|E|$
- Tree: $e = v - 1$
- Planar connected: $v - e + f = 2$
- Simple planar: $e \leq 3v - 6$
- Triangle-free planar: $e \leq 2v - 4$
- Coloring: $\omega(G) \leq \chi(G) \leq \Delta(G) + 1$
- Full m-ary tree: $n = mi + 1, \ell = (m-1)i + 1$

15.11 Boolean and Computation / 布尔与计算

- DNF: OR of minterms for rows where output is 1
- CNF: AND of maxterms for rows where output is 0
- NAND: $\neg x = x \uparrow x$
- Halting problem is undecidable.

16. Quiz 高频题型 / Frequent Quiz Patterns

16.1 逻辑题 / Logic

- 判断命题是否等价：真值表或等价律。
- 否定含量词命题：翻转量词并把否定推进谓词。
- 证明 implication：优先考虑直接证明或逆否证明。

16.2 数论题 / Number Theory

- 求 gcd：欧几里得算法。
- 求模逆：扩展欧几里得。
- 解线性同余：先看 gcd 是否整除右边。
- 大幂取模：费马、欧拉、CRT、快速幂。

16.3 计数题 / Counting

- 是否有顺序：排列 vs 组合。
- 是否允许重复：星棒法或重复排列。
- 是否有“至少一个坏事件”：容斥或补集。
- 是否是“没有固定点”：错排。

16.4 概率题 / Probability

- 等可能样本空间：计数。
- 已知条件：条件概率。
- 反向条件：Bayes。
- 随机计数：指示变量与期望线性性。

16.5 图论题 / Graph Theory

- 一笔画：欧拉路 / 欧拉回路。
- 每点一次：哈密顿路 / 哈密顿圈。
- 平面性：Euler 公式和边数上界。
- 二分图：无奇圈或 BFS 两色。
- 树：连通无圈或 $e = v - 1$。

17. 中英术语表 / Bilingual Glossary

- proposition: 命题
- predicate: 谓词
- quantifier: 量词
- implication: 蕴含
- contrapositive: 逆否命题
- contradiction: 矛盾 / 反证
- set: 集合
- subset: 子集
- power set: 幂集
- function: 函数
- injective function / injection: 单射
- surjective function / surjection: 满射
- bijective function / bijection: 双射
- countable: 可数

- uncountable: 不可数
- divisibility: 整除
- congruence: 同余
- modular inverse: 模逆元
- Chinese remainder theorem: 中国剩余定理
- induction: 归纳法
- recurrence: 递推关系
- generating function: 生成函数
- permutation: 排列
- combination: 组合
- pigeonhole principle: 鸽巢原理
- derangement: 错排
- random variable: 随机变量
- expectation: 期望
- variance: 方差
- equivalence relation: 等价关系
- partial order: 偏序
- graph: 图
- tree: 树
- planar graph: 平面图
- matching: 匹配
- Boolean algebra: 布尔代数
- finite automaton: 有限自动机
- Turing machine: 图灵机