离散数学笔记

ixtWuko

2018 年 4 月 8 日

1 逻辑、证明

1.1 命题与逻辑

- 什么是命题？或真或假的陈述语句。
- $\neg p$: NOT p　$p \wedge q : p$ AND q　$p \vee q : p$ OR q　$p \oplus q : p$ XOR q
- Condition Statement (条件语句) $p \to q$: if p is true than q is true.
- Bicondition Statement $p \leftrightarrow q$. 当 $p \leftrightarrow q$ 恒真，则 p, q 等价 ($\equiv$)。
- 永真式（永远为真的复合命题）、矛盾式（永远为假的复合命题）、可能式
- 命题函数：设命题 $P(x)$ 为一个关于变量 x 的函数，则称 $P(x)$ 为命题函数，其中 P 称为谓语。
- 全称量词 $\forall$、存在量词 $\exists$

常用逻辑等式

- $p \wedge T \equiv p$　$p \vee F \equiv p$　$p \vee T \equiv T$　$p \wedge F \equiv F$
- $p \vee p \equiv p$　$p \wedge p \equiv p$　$\neg(\neg p) \equiv p$
- $p \vee q \equiv q \vee p$　$p \wedge q \equiv q \wedge p$
- $(p \vee q) \vee r \equiv p \vee (q \vee r)$　$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$
- $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$　$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$
- $\neg(p \wedge q) \equiv \neg p \vee \neg q$　$\neg(p \vee q) \equiv \neg p \wedge \neg q$
- $p \vee (p \wedge q) \equiv p$　$p \wedge (p \vee q) \equiv p$
- $p \vee \neg p \equiv T$　$p \wedge \neg p \equiv F$
- $p \to q \equiv \neg q \to \neg p$
- $p \leftrightarrow q \equiv (p \to q) \wedge (q \to p)$

---

[0] 模糊逻辑：模糊逻辑常用于人工只能，其命题的真值在 0 和 1 之间，表示不同的程度。

- 模糊逻辑中命题的反是 1 减命题的真值。
- 模糊逻辑中命题的交是所有命题真值的最小值。
- 模糊逻辑中命题的并是所有命题真值的最大值。

1.2 证明

证明方法和技巧

- 直接证明、间接证明（反证法）
- 利用矛盾证明（归谬证明）、利用等价证明
- 存在性证明、唯一性证明
- 穷举证明、数学归纳法、结构归纳法
- 康托尔对角化方法、组合证明

开放问题

- 费马大定理：$n > 2, \neg\exists x, y, z \in \mathbb{Z}, that\ x^n + y^n = z^n$.（费马大定理已证明）
- $3x+1$ 猜想：设 $T(x) = \begin{cases} \frac{x}{2} & \text{x is even} \\ 3x+1 & \text{x is odd} \end{cases}$，对于所有的正整数，重复使用 $T(x)$，最终结果必为 1.

2 集合、函数、数列

2.1 集合

- $\mathbb{N}$：自然数　$\mathbb{Z}$：整数　$\mathbb{Z}^+$：正整数　$\mathbb{Q}$：有理数　$\mathbb{R}$：实数
- 集合是无序的。
- element $\in$ set, subset $\subseteq$ set. $\emptyset \subseteq S, S \subseteq S$
- 集合的基数指集合中的不同元素的个数，记作 $|S|$。若 $|S| \to \infty$，则集合 S 为无限集合，否则为有限集合。
- 集合的幂：集合 S 的幂为 S 所有子集的集合，记作 $P(S)$。
  如 $P(\{1,2\}) = \{\emptyset, \{1\}, \{2\}, \{1,2\}\}$.
- 有序 n 元组：有序 n 元组 $(a_1, a_2, \cdots, a_n)$ 是有顺序的。当 $n = 2$ 时，称为有序二元组。
- 笛卡儿积：$A \times B = \{(a,b) \mid a \in A \wedge b \in B\}$
  $A_1 \times A_2 \times \cdots \times A_n = \{(a_1, a_2, \cdots, a_n) \mid a_i \in A_i\}$
- 集合基本运算：$A \cap B$　$A \cup B$　$\bigcap_{i=1}^{n} A_i$　$\bigcup_{i=1}^{n} A_i$　$A - B$　$\overline{A}$.

- 一个有限集合或基数等于正整数集的集合，称为可数集合；否则称为不可数集合。（无限集合可能是可数的，只要基数等于正整数集即可。）
  正整数集的基数记为 $\aleph_0$，即 $|\mathbb{Z}^+| = \aleph_0$。(aleph null，阿列夫零)

常用集合等式

- $A \cup (B \cup C) = (A \cup B) \cup C \quad A \cap (B \cap C) = (A \cap B) \cap C$
- $A \cap (B \cup C) = (A \cap B) \cup (A \cap C) \quad A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
- $\overline{A \cup B} = \overline{A} \cap \overline{B} \quad \overline{A \cap B} = \overline{A} \cup \overline{B}$
- $A \cup \overline{A} = U \quad A \cap \overline{A} = \emptyset$

2.2 函数

- 若 f_1, f_2 均是从 A 到 $\mathbb{R}$ 的函数，则 $f_1 + f_2, f_1 f_2$ 也是从 A 到 $\mathbb{R}$ 的函数，且

$$(f_1 + f_2)(x) = f_1(x) + f_2(x)$$

$$(f_1 f_2)(x) = f_1(x) f_2(x)$$

- 若 f 是从 A 到 B 的函数，$S \subset A$，则 $f(S) = \{f(s) \mid s \in S\}$。
- 若 f 是从 A 到 B 的函数，对于集合 B 中的任何一个元素 b，都有且仅有一个属于集合 A 的元素与之对应，使得 $f(a) = b$，则称 f 为一一对应或双射。
- 若 f 是从 A 到 B 的一一对应的函数，则存在 f 的反函数 f^{-1}。
- 若 g 是从 A 到 B 的函数，f 是从 B 到 C 的函数，则记

$$f \circ g = f(g(x))$$

- floor function: $\lfloor x \rfloor$ or $[x] \quad$ ceiling function: $\lceil x \rceil$

[0]模糊集合：全集 U 中的每个元素在集合 S 中都有成员度，处于 0 和 1 之间。

- 模糊集合的补集的元素的成员度为 1 减去该元素在原集合中的成员度。
- 模糊集合的并集是取元素在各集合中成员度的最大值。
- 模糊集合的交集是取元素在各集合中成员度的最小值。

2.3 数列

- 斐波那契数列：$f_0 = 0, f_1 = 1, f_n = f_{n-1} + f_{n-1}$
  一对刚出生的兔子放在一个食物充足的孤岛上，假设兔子 2 个月才能长大成熟，每个月每一对成熟的兔子会繁殖出一对新的兔子，并且兔子不会死去，兔子的对数满足斐波那契数列。
- 几何数列 ar^k 的前 n 项和：$S = \dfrac{a(r^n - 1)}{r - 1}$.
- $\displaystyle\sum_{k=1}^{n} k = \frac{n(n+1)}{2}$
- $\displaystyle\sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}$
- $\displaystyle\sum_{k=1}^{n} k^3 = \frac{n^2(n+1)^2}{4}$
- $\displaystyle\sum_{k=0}^{\infty} x^k (|x| < 1) = \frac{1}{1-x}$
- $\displaystyle\sum_{k=1}^{\infty} kx^{k-1} (|x| < 1) = \frac{1}{(1-x)^2}$

3 算法、整数、矩阵

3.1 算法

- 常见算法：顺序搜索、二分搜索、冒泡排序、插入排序 …
- 贪心算法：每一步都寻找局部最优解的算法，最后的结果不一定是最优。例如凑硬币的问题使用贪心算法，但是最后所得解即为最优解。

停机问题

停机问题：是否存在一个程序，它的输入是某个计算机程序，输出是这个计算机程序能否正常结束。这样的程序是不存在的。

证明. 构造一个程序 $H(P, I)$，其中 P 为一个计算机程序，I 为 P 的输入；其输出为 P 能否正常结束，当 P 停机时，输出字符串“停机”，当 P 死循环时，输出字符串“死循环”。

因为计算机程序 P 的源代码同时也是文本，同样可以作为 P 的输入。假设另一程序 $K(P)$，其运行结果与 $H(P, P)$ 相反，当 $H(P, P)$ 输出为“死循环”，那么 $K(P)$ 停机，当 $H(P, P)$ 输出为“停机”，那么 $K(P)$ 死循环。

此时考虑 $H(K,K)$，如果 $H(K,K)$ 输出为“死循环”，也即 $K(K)$ 死循环，但是由上述 $K(P)$ 的功能，可知 $K(P)$ 停机。两种结果相矛盾，$H(P,P)$ 不能得到正确的结果，因此不存在这样的程序。 □

大 O 记法

- 存在常数 C,k 使得当 $x>k$ 时，$|f(x)|\le C|g(x)|$，则称 $f(x)$ 是 $O(g(x))$.
- $f(x)=a_nx^n+a_{n-1}x^{n-1}+\cdots+a_1x+a_0$，$f(x)$ 是 $O(x^n)$.
- $1+2+3+\cdots+n$ 是 $O(n^2)$，$n!$ 是 $O(n^n)$，$\log n$ 是 $O(n)$，$\log(n!)$ 是 $O(n\log n)$.
- 常用的有 $1,\log n,n,n\log n,n^2,2^n,n^n$.
- 若 $f_1(x)$ 是 $O(g_1(x))$，$f_2(x)$ 是 $O(g_2(x))$，则 $(f_1+f_2)(x)$ 是 $O(\max(|g_1(x)|,|g_2(x)|))$，$(f_1f_2)(x)$ 是 $O(g_1(x)g_2(x))$.
- 存在常数 C,k 使得当 $x>k$ 时，$|f(x)|\ge C|g(x)|$，则称 $f(x)$ 是 $\Omega(g(x))$.
- 若 $f(x)$ 是 $O(g(x))$ 且 $f(x)$ 是 $\Omega(g(x))$，则 $f(x)$ 是 $\Theta(g(x))$，或称 $f(x)$ 是 $g(x)$ 阶的。
- $f(x)=a_nx^n+a_{n-1}x^{n-1}+\cdots+a_1x+a_0$，$f(x)$ 是 x^n 阶的。

算法复杂度

- 常数复杂度：$\Theta(1)$　对数复杂度 $\Theta(\log n)$　线性复杂度 $\Theta(n)$
  多项式复杂度 $\Theta(n^a)$　指数复杂度 $\Theta(b^n)$　阶乘复杂度 $\Theta(n!)$
- 最坏情形复杂度、平均情形复杂度
- NP 类问题：在多项式复杂度内无法解决，但是可在多项式复杂度内验证答案是否正确。
  P 类问题：易解问题，可在多项式复杂度内解决的。

3.2 整数

整除与余数

a 整除 b：记作 $a|b$，也即 b/a 为整数。$a=dq+r$，则 $q=a\ \mathrm{div}\ d, r=a\bmod d$.

- 若 $a|b,a|c$，则 $a|(mb+nc)$.
- 若 $a|b$，则 $\forall c\in\mathbb{Z},a|bc$.
- 若 $a|b,b|c$，则 $a|c$.

若 m 能整除 $a-b$，则 $a \bmod m = b \bmod m$，此时 a, b 同余，记作 $a \equiv b \pmod m$，且存在整数 k 使得 $a = b + km$.

- 若 $a \equiv b \pmod m, c \equiv d \pmod m$，则 $a + c \equiv b + d \pmod m, ac = bd \pmod m$.
- $(a + b) \bmod m = ((a \bmod m) + (b \bmod m)) \bmod m$
  $ab \bmod m = ((a \bmod m)(b \bmod m)) \bmod m$

  同余的应用：
- 哈希函数：$h(k) = k \bmod m$，其中 k 为记录的 key。哈希值与 key 并不是一一对应的。
- 线性同余法生成伪随机数：选定 4 个整数 $a, c, m, x_0, 2 \le a < m, 0 \le c < m, 0 \le x_0 < m$，其中 x_0 称为种子。使用 $x_{n+1} = (ax_n + c) \pmod m$ 生成一个伪随机数序列。
- 数字校验中的奇偶校验：对于数字序列 $x_1x_2 \cdots x_n$，奇偶校验位 $x_{n+1} = (x_1 + x_2 + \cdots + x_n) \bmod 2$.
  其它的如通用产品代码 UPC，也即商品包装上的条形码；国际标准书号 ISBN 等。

素数

- 算数基本定理：任何大于 1 的正整数，都可以唯一地写成两个或多个素数的乘积。
- 若 n 为合数，则 n 必有小于或等于 $\sqrt{n}$ 的素因子。
- 有无限多的素数。
- 梅森素数：若 p 为素数，且 $2^p - 1$ 也为素数，则称 $2^p - 1$ 为梅森素数。
- 素数定理：当 x 无限增长时，不超过 x 的素数的个数与 $x/\ln x$ 之比趋向于 1。
- 哥德巴赫猜想：任何一个大于 2 的奇数，必为 2 个素数之和。
- 孪生素数猜想：有无限多的孪生素数。

最大公约数和最小公倍数

最大公约数 $\gcd(a, b)$，最小公倍数 $\operatorname{lcm}(a, b)$.

- $ab = \gcd(a, b) \cdot \operatorname{lcm}(a, b)$
- 使用欧几里得算法求最大公约数：若 $a, b, q, r \in \mathbb{Z}, a = dq + r$，则 $\gcd(a, b) = \gcd(b, r)$，辗转相除，求得最大公约数。

- 贝祖等式：若 $a,b\in\mathbb{Z}^+$，则 $\exists s,t\in\mathbb{Z},gcd(a,b)=sa+tb$（$s,t$ 可能不全为正）。
- 若 $a,b,c\in\mathbb{Z}^+,\gcd(a,b)=1,a|bc$，则 $a|c$.
- p 为素数，$p|a_1a_2\cdots a_n$，则 $\exists a_i\in\{a_1,a_2,\cdots,a_n\},p|a_i$.
- 若 $m\in\mathbb{Z}^+,a,b,c\in\mathbb{Z},ac\equiv bc\pmod m,\gcd(c,m)=1$，则 $a\equiv b\pmod m$.

相关问题

- 线性同余方程：$ax\equiv b\pmod m$，解为 $x_0+k\frac{m}{d},d=\gcd(a,n)$.
  解法：利用 $a\overline{a}\equiv 1\pmod m$，其中 $\overline{a}$ 是 a 的模 m 的逆，方程两侧同乘 $\overline{a}$，即可解得。
- 若 $a,m\in\mathbb{Z}$ 互素，$m>1$，则存在 a 的模 m 的逆，且该逆模 m 是唯一的。
  例如：$a=8,m=3$，则 $\overline{a}=2+3k,\overline{a}\bmod 3=2$.
- 中国剩余定理：若 $m_1,m_2,\cdots,m_n$ 为两两互素的素数，则

$$\begin{aligned}x&\equiv a_1\pmod{m_1}\\x&\equiv a_2\pmod{m_2}\\&\cdots\\x&\equiv a_n\pmod{m_n}\end{aligned}$$

  有唯一的模 m 的解，其中 $m=m_1m_2\cdots m_n$。可以使用此定理进行大整数的运算。

  例如：取 m 为 99,98,97,95，123684 可以表示为 (33,8,9,89)，413456 可以表示为 (32,92,42,16)，则 123684+413456 可以使用如下计算：

$$\begin{aligned}&(33,8,9,89)+(32,92,42,16)\\&\quad=(65\bmod 99,100\bmod 98,51\bmod 97,105\bmod 95)\\&\quad=(65,2,51,10)\end{aligned}$$

  根据中国剩余定理，

$$\begin{aligned}x&=65\pmod{99}\\x&=2\pmod{98}\\x&=51\pmod{97}\\x&=10\pmod{95}\end{aligned}$$

  可得 537140 是小于 $99\times98\times97\times95$ 的惟一解。因此，可以使用中国剩余定理进行大整数运算。

- 费马小定理：p 为素数，a 是不能被 p 整除的整数，则 $a^{p-1} \equiv 1 \pmod{p}$. 因而对于所有的整数 a，都有 $a^p \equiv p \pmod{p}$.
- 伪素数：b 为正整数，n 为正合数，且 $b^{n-1} \equiv 1 \pmod{n}$，则 n 称为基数为 b 的伪素数。
- 卡米切尔数：b 为正整数，n 为正合数，且 $\gcd(b,n) = 1, b^{n-1} \equiv 1 \pmod{n}$，则称 n 卡米切尔数。
- 原根：整数 r 和素数 p，若 $1 < r < p, 0 < i < p$，且 $r^i \bmod p$ 两两不同（有 $p-1$ 种结果），则称 r 为 p 的一个原根。也即 $[1, p-1]$ 上的所有整数 i，当且仅当 $i = p-1$ 时，$r^i \equiv 1 \pmod{p}$.

密码

- 密码系统：包含 5 部分，明文 $\mathcal{P}$、密文 $\mathcal{C}$、密钥空间 $\mathcal{K}$、加密函数 $\mathcal{E}$、解密函数 $\mathcal{D}$。若密钥 k 对应的加密函数为 E_k，揭秘函数为 D_k，则 $D_k(E_k(p)) = p$.
- 经典密码学：
  - 凯撒密码
  - 分组密码：将文本分成一定长度的单元，每个单元按照一定的方式加密。
  - 转置密码：设函数 $\sigma([1, 2, \cdots, n]) = [m_1, m_2, \cdots, m_n]$，将文本分成长度为 n 的分组，按 $1, 2, \cdots, n$ 的位置，分别移位 $m_1, m_2, \cdots, m_n$。
- 公钥密码学：加密密钥是公开的，解密密钥是保密的。

  例如，RSA 密码系统：加密需要两个整数 n, e，其中公钥 $n = pq$ 为两个大素数的乘积，p, q 各约 200 位，e 为与 $(p-1)(q-1)$ 互素的指数。加密过程首先将文本翻译成整数，然后分组，每一组为一个大整数 M；密文；$C = M^e \bmod n$.

  解密需要解密密钥 d，d 为 e 模 $(p-1)(q-1)$ 的逆。也即 $de \equiv 1 \pmod{(p-1)(q-1)} \to de = 1 + k(p-1)(q-1)$.

  因此 $C^d \equiv (M^e)^d \equiv M^{de} \equiv M^{1+k(p-1)(q-1)} \pmod{n}$。假定 $\gcd(M, p) = \gcd(M, q) = 1$（这一关系不成立的可能性很小），则根据费马小定理

  $C^d \equiv M \cdot (M^{p-1})^{k(q-1)} \equiv M \cdot 1 \equiv M \pmod{p}$
  $C^d \equiv M \cdot (M^{q-1})^{k(p-1)} \equiv M \cdot 1 \equiv M \pmod{q}$.

  根据中国剩余定理，$C^d \equiv M \pmod{pq}$.

3.3 矩阵

对于矩阵 $\mathbf{A},\mathbf{B}$，

- $\mathbf{A}+\mathbf{B}=[\mathbf{a_{ij}}+\mathbf{b_{ij}}]$
- $\mathbf{AB}=[\mathbf{c_{ij}}],\mathbf{c_{ij}}=\mathbf{a_{i1}b_{1j}}+\mathbf{a_{i2}b_{2j}}+\cdots+\mathbf{a_{ik}b_{kj}}$
- n 阶单位阵 $\mathbf{I_n}$，矩阵的幂 $\mathbf{A^0}=\mathbf{I_n}\quad \mathbf{A^r}=\underbrace{\mathbf{AA}\cdots\mathbf{A}}_{\text{r 次}}$
- 转置 $\mathbf{A^t}$，对称阵 $\mathbf{A^t}=\mathbf{A}$
- 0-1 矩阵，元素只有 0 和 1 的矩阵。对于 0-1 矩阵：
  - $\mathbf{A}\wedge\mathbf{B}=[\mathbf{a_{ij}}\wedge\mathbf{b_{ij}}]$，$\mathbf{A}\vee\mathbf{B}=[\mathbf{a_{ij}}\vee\mathbf{b_{ij}}]$
  - 布尔积 $\mathbf{A}\odot\mathbf{B}=[\mathbf{c_{ij}}],\mathbf{c_{ij}}=(\mathbf{a_{i1}}\wedge\mathbf{b_{1j}})\vee(\mathbf{a_{i2}}\wedge\mathbf{b_{2j}})\vee\cdots\vee(\mathbf{a_{ik}}\wedge\mathbf{b_{kj}})$
  - 布尔幂 $\mathbf{A^{[r]}}=\underbrace{\mathbf{A}\odot\mathbf{A}\odot\mathbf{A}\odot\cdots\odot\mathbf{A}}_{\text{r 次}}$

4 归纳与递归

4.1 归纳

- 数学归纳法：(1)$P(1)$ 为真；(2) 假设 $P(k)$ 为真，可以推得 $P(k+1)$ 为真。由两步可以证得 $\forall kP(k)$ 成立。
- 强归纳法：(1)$P(1)$ 为真；(2) 假设 $P(1),P(2),\cdots,P(k)$ 全为真，可以推得 $P(k+1)$ 为真。由两步可以证得 $\forall kP(k)$ 成立。也即，证明 $P(k+1)$ 为真不仅需要前一项，而需要之前的所有项。
- 尽量使用强归纳法，强归纳法是完全的归纳法。
- 数学归纳法和强归纳法的有效性都来自于良序性公理：任何一个非空的非负整数集合都有最小元素。
- 奇数个馅饼问题：奇数个人站在一个院子里，彼此间距离不同，他们同时将一个馅饼扔向离自己最近的人，至少有一个人没有被扔到。（使用上述的良序性公理可证）
- 具有 n 条边的简单多边形可以三角化成 $n-2$ 个三角形，其中 $n>3$.
- 结构归纳法：(1) 对于递归定义中的基础步骤所规定的所有元素，结论成立；(2) 对于递归步骤中，假设用于构造新元素的所有元素结论成立，则对于新元素结论成立。由以上两步可以证得结论成立。

4.2 递归

利用递归来定义、利用递归来求解。

4.3 程序正确性

证明一个程序的正确性分为两个部分：(1) 若程序终止，则必定获得正确的答案；(2) 程序总是终止。前一部分称为程序的部分正确性。

若每当对程序段 S 来说，初始断言 p 为真时，其终结断言 q 必为真，则称程序段 S 相对于 p 和 q 部分正确，以 $p\{S\}q$ 表示程序部分正确。

5 离散概率

概率的内容看概率论与数理统计部分。

随机算法可以大致分为两类：（不是遍历的算法，也即随机采样而不能覆盖所有情况。）

- 蒙特卡罗算法：采样越多，越近似最优解。尽量找好的，但是不能保找到的是最好的。
- 拉斯维加斯算法：采样越多，越有机会找到最优解。尽量找最好的，但是不能保证找到。

6 计数

- 基本计数原则：乘积原则和求和原则。乘积原则是指完成一个任务的多个步骤，分别有多种不同的方法，则完成此任务的方法为各步骤中方法数量的乘积。求和原则是指某个事物存在多种情况，情况之间需要求和。
- 容斥原理：求和时将某些情况重复计数，需要减去重复的。
- 鸽巢原理/抽屉原理：如果 $k+1$ 个或更多的物体放入 k 个盒子，则至少有一个盒子包含 2 个或更多的物体。

  广义鸽巢原理：如果 N 个物体放入 k 个盒子，则至少有一个盒子包含了至少 $\lceil N/k \rceil$ 个物体。
- 具有 n 个不同元素的集合的 r 排列数是

$$P(n,r) = n(n-1)(n-2)\cdots(n-r+1) = \frac{n!}{(n-r)!}$$

.

- 具有 n 个不同元素的集合的 r 组合数是

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

.

  - $\binom{n}{r}=\binom{n}{n-r}$
  - 帕斯卡恒等式：$\binom{n+1}{r}=\binom{n}{r-1}+\binom{n}{r}$
  - 范德蒙德恒等式：$\binom{m+n}{r}=\sum_{k=0}^{r}\binom{m}{r-k}\binom{n}{k}$；令 $m=r=n$，则 $\binom{2n}{n}=\sum_{k=0}^{n}\binom{n}{k}^2$.
  - $\binom{n+1}{r+1}=\sum_{j=r}^{n}\binom{j}{r}$

- 二项式定理：

$$(x+y)^n=\sum_{r=0}^{n}\binom{n}{r}x^{n-r}y^r$$

  - 令 $x=y=1$，则 $\sum_{r=0}^{n}\binom{n}{r}=2^n$.
  - 令 $x=-1,y=1$，则 $\sum_{r=0}^{n}(-1)^r\binom{n}{r}=0$.
  - 令 $x=2,y=1$，则 $\sum_{r=0}^{n}\binom{n}{r}2^r=3^n$.

- 具有 n 个物体的集合允许重复的 r 排列数是 n^r.
  具有 n 个物体的集合允许重复的 r 组合数是 $\binom{n+r-1}{r}=\binom{n+r-1}{n-1}$.
- 设类型 1 的相同的物体有 n_1 个，类型 2 的相同的物体有 n_2 个，...，类型 k 的相同的物体有 n_k 个，那么 n 个物体的不同排列数是 $\frac{n!}{n_1!n_2!\cdots n_k}$.
- 把 n 个不同的物体分配到 k 个不同的盒子使得 n_i 个物体放入盒子 i 的方式数等于 $\frac{n!}{n_1!n_2!\cdots n_k}$.

7 高级计数技术

7.1 递推关系

序列的递推关系是一个等式，用前面的一项或多项来描述第 n 项 a_n。满足递推关系的一个序列称为该递推关系的解。

常系数线性齐次递推关系

常系数的 k 阶线性齐次递推关系：

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

其中 $c_1, c_2, \cdots, c_k$ 为实数，$c_k \neq 0$。
假设 $a_n = r^n$，则上式可以写成

$$r^n = c_1 r^{n-1} + c_2 r^{n-2} + \cdots + c_k r^{n-k}$$

两侧同除以 r^{n-k}，得

$$r^k - c_1 r^{k-1} - c_2 r^{k-2} - \cdots - c_{k-1} r - c_k = 0$$

上式称为递推关系的特征方程，此时需要求解特征根 r.

- 若有 k 个不同的特征根 $r_1, r_2, \cdots, r_k$，解为

$$a_n = \alpha_1 r_1^n + \alpha_2 r_2^n + \cdots + \alpha_k r_k^n$$

其中 $\alpha_1, \alpha_2, \cdots, \alpha_k$ 为常数，使用初始条件解出 $\alpha_1, \alpha_2, \cdots, \alpha_k$.

- 若有 t 个不同的特征根 $r_1, r_2, \cdots, r_t$，其重数分别为 $m_1, m_2, \cdots, m_t$，解为

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

其中所有的 α 为常数，由初始条件解出。

以二阶为例，特征方程为 $r^2 - c_1 r - c_2 = 0$，解得特征根：

- 若 $r_1 \neq r_2$，则 $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$.
- 若 $r_1 = r_2 = r_0$，则 $a_n = (\alpha_1 + \alpha_2 n) r_0^n$.

常系数线性非齐次递推关系

常系数的 k 阶线性非齐次递推关系：

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

其中 $c_1, c_2, \cdots, c_k$ 为实数，$c_k \neq 0$.

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

其中 $b_0, b_1, \cdots, b_t, s$ 为实数。
其解的结构是特解 $a_n^{(p)}$+ 通解 $a_n^{(h)}$，其中通解为对应的齐次递推关系的解。
下面求其特解：

- 若 s 不是特征根，则 $a_n^{(p)} = (p_1 n^t + p_{t-1} n^{t-1} + \cdots + p_1 n + p_0) s^n$。
- 若 s 是一个 m 重特征根，则 $a_n^{(p)} = n^m (p_1 n^t + p_{t-1} n^{t-1} + \cdots + p_1 n + p_0) s^n$.

将以上的特解形式带入递推关系，化简并归并同类项，根据不同次的项的系数对应相同，解得以上的各 p.

分治递推关系

分治递推关系：

$$f(n) = af(n/b) + g(n)$$

- 设 $f(n) = af(n/b) + c$，且为增函数，其中 b 整除 n，$a \geq 1, b \geq 1, b \in \mathbb{Z}, c > 0$。则若 $a = 1$，$f(n)$ 是 $O(\log n)$；若 $a > 1$，$f(n)$ 是 $O(n^{\log_b a})$.

  进一步，当 $n = b^k, k > 0, k \in \mathbb{Z}$，$f(n) = C_1 n^{\log_b a} + C_2$，其中 $C_1 = f(1) + \dfrac{c}{a-1}, C_2 = -\dfrac{c}{a-1}$.
- 主定理 设 $f(n) = af(n/b) + cn^d$，且为增函数，其中 $n = b^k$，$a \geq 1, b \geq 1, b \in \mathbb{Z}$，$c > 0, d \geq 0$。则若 $a < b^d$，$f(n)$ 是 $O(n^d)$；若 $a = b^d$，$f(n)$ 是 $O(n^d \log n)$；若 $a > b^d$，$f(n)$ 是 $O(n^{\log_b a})$.

上述内容可以用于估算含有递归的程序的运行时间，尤其是主定理或更广泛的主定理。

7.2 生成函数

实数序列 $\{a_0, a_1, \cdots, a_k, \cdots\}$ 的生成函数是无穷级数 $G(x) = \sum_{k=0}^{\infty} a_k x^k$，生成函数的各项的系数为序列的元素。

在计数问题中，通常考虑使用幂级数，下面是一些有用结论：

- 设 $f(x) = \sum_{k=0}^{\infty} a_k x^k, g(x) = \sum_{k=0}^{\infty} b_k x^k$，则

$$f(x) + g(x) = \sum_{k=0}^{\infty} (a_k + b_k) x^k$$

$$f(x) g(x) = \sum_{k=0}^{\infty} (\sum_{j=0}^{k} a_j b_{k-j}) x^k$$

- 广义二项式系数：

$$\binom{u}{k} = \begin{cases} \dfrac{u(u-1)(u-2)\cdots(u-k+1)}{k!} & k > 0 \\ 1 & k = 0 \end{cases}$$

其中 u 为实数，k 为非负整数。

- 广义二项式定理：

$$(1+x)^u = \sum_{k=0}^{\infty} \binom{u}{k} x^k$$

其中 $|x| < 1$，u 为实数。

使用生成函数解决计数问题的例题：

将 8 块饼干分给 3 个孩子，每个孩子不少于 2 块不多于 4 块，问有多少种分法？

分析：需要寻找 3 个整数，其和为 8。以幂级数考虑，可以使用多项式的乘积，其指数为加和。

解：使用多项式 $x^2+x^3+x^4$ 来表示每个孩子可能分得的饼干数量，而 3 个孩子分得饼干的数量总和可以使用生成函数 $(x^2+x^3+x^4)^3$ 来获得。其中，x^8 的系数即为本题的解。

使用生成函数解决递推关系的例题：

求解递推关系 $a_k = 3a_{k-1}$，初始条件 $a_0 = 2$.

解：设序列 $\{a_n\}$ 的生成函数 $G(x) = \sum_{k=0}^{\infty} a_k x^k$，

$$xG(x) = \sum_{k=0}^{\infty} a_k x^{k+1} = \sum_{k=1}^{\infty} a_{k-1} x^k$$

$$\begin{aligned} G(x) - 3xG(x) =& \sum_{k=0}^{\infty} a_k x^k - 3\sum_{k=1}^{\infty} a_{k-1} x^k \\ =& a_0 + \sum_{k=1}^{\infty} (a_k - 3a_{k-1}) x^k \\ =& 2 \end{aligned}$$

$$G(x) = \frac{2}{1-3x} = \sum_{k=0}^{\infty} 2 \times 3^k x^k$$

所以，$a_n = 2 \times 3^n$.

7.3 容斥

容斥原理：设 $A_1, A_2, \cdots, A_n$ 是有穷集，则

$$\begin{aligned} |A_1 \cup A_2 \cup \cdots \cup A_n| = & \sum_{1 \le i \le n} |A_i| - \sum_{1 \le i < j \le n} |A_i \cap A_j| \\ & + \sum_{1 \le i < j < k \le n} |A_i \cap A_j \cap A_k| - \cdots + (-1)^{n+1} |A_1 \cap A_2 \cap \cdots \cap A_n| \end{aligned}$$

容斥原理的另一种表述：设 A_i 表示集合中含有性质 P_i 的元素构成的子集，使用 $N(P_1, P_2, \cdots, P_k)$ 表示具有 $P_1, P_2, \cdots, P_k$ 所有这些性质的元素的数量。有

$$|A_1 \cap A_2 \cap \cdots \cap A_k| = N(P_1, P_2, \cdots, P_k)$$

不具有 $P_1, P_2, \cdots, P_k$ 中任何一条性质的元素的数量用 $N(P_1', P_2', \cdots, P_k')$ 表示。
由容斥原理是，可得

$$\begin{aligned} N(P_1', P_2', \cdots, P_n') =& N - |A_1 \cup A_2 \cup \cdots \cup A_n| \\ =& N - \sum_{1 \le i \le n} N(P_i) + \sum_{1 \le i < j \le k} N(P_i P_j) - \sum_{1 \le i < j < k \le n} N(P_i P_j P_k) \\ & + \cdots + (-1)^n N(P_1 P_2 \cdots P_n) \end{aligned}$$

容斥原理的应用

- 埃拉托色妮筛：寻找不超过一个给定正整数的素数。一次去掉被 2 整除的数，被 3 整除的数，被 5 整除的数，…

  求这些素数的个数，可以使用容斥原理。

- 求两个集合间的映上函数的数量：$m, n \in \mathbb{Z}^+, m \ge n$，则存在 $n^m - C(n,1)(n-1)^m + C(n,2)(n-2)^m - \cdots + (-1)^{n-1} C(n, n-1) \times 1^m \cdot 1^m$ 个从 m 元素集合到 n 元素集合的映上函数。

- 错位排列指没有一个元素处于初始位置的排列。

  n 元素集合的错位排列数是 $D_n = n!\left[1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} + \cdots + (-1)^n \frac{1}{n!}\right]$.

8 关系

8.1 二元关系

一个从集合 A 到集合 B 的二元关系是有序对的集合 R，其中每个有序对的第一个元素取自 A，第二个元素取自 B。

使用 aRb 表示 $(a,b) \in R$，称 a 与 b 有关系 R.

集合 A 的关系是从 A 到 A 的关系。

例如：$A = \{1,2,3,4,5\}$，A 上的关系 $R = \{(a,b) \mid a$整除$b\}$ 中有哪些有序对？

二元关系的性质

- 一个从集合 A 到集合 B 的二元关系是 $A \times B$ 的子集。

- 若对于每个元素 $a \in A$ 有 $(a,a) \in R$，则称集合 A 上的关系 R 是自反的。

- 对于元素 $a, b \in A$，若只要 $(a,b) \in R$ 就有 $(b,a) \in R$，则称集合 A 上的关系 R 是对称的。

对于元素 $a,b \in A$，仅当 $a=b$ 时，有 $(a,b) \in R$ 和 $(b,a) \in R$，则称集合 A 上的关系 R 是反对称的。

- 对于元素 $a,b,c \in A$，若 $(a,b) \in R$ 且 $(b,c) \in R$，则 $(a,c) \in R$，则称集合 A 上的关系 R 是传递的。
- 设 R 是从集合 A 到集合 B 的关系，S 是从集合 B 到集合 C 的关系，R 和 S 的合成是由有序对 (a,c) 构成的关系，其中 $a \in A, c \in C$，并且对于它们存在一个元素 $b \in B$ 使得 $(a,b) \in R, (b,c) \in S$。用 $S \circ R$ 表示 R 和 S 的合成。
- 设 R 是集合 A 上的关系，则幂 R^n 递归地定义为 $R^1 = R, R^{n+1} = R^n \cdot R$.

  当且仅当对于 $n = 1,2,3$ 有 $R^n \subset R$ 时，集合 A 上的关系 R 是传递的。

二元关系的表示

二元关系的表示可以使用矩阵、有向图。

使用矩阵表示二元关系，矩阵中的元素的位置代表了集合中的不同的元素，矩阵中的元素的值为 1 或 0，表示该元素对应的集合中的两个元素之间有无关系。

使用有向图表示二元关系，每个节点代表集合中的一个元素，节点之间的有向线段表示元素之间的关系，如有从节点 a 指向节点 b 的线段，则表示存在关系 (a,b).

8.2 n 元关系

在集合 $A_1, A_2, \cdots, A_n$ 上的 n 元关系是 $A_1 \times A_2 \times \cdots \times A_n$ 的子集。其中 $A_1, A_2, \cdots, A_n$ 称为关系的域，n 叫做关系的阶。

例如，关系数据类型中的每个字段都是一个 n 元组，一条记录就是就是一个 n 元关系。

以下是 n 元关系的一些运算：

- 选择运算：设 R 是 n 元关系，C 是 R 中元素可能满足的条件，则选择运算 s_c 是将 R 映射到 R 中满足 C 的所有 n 元组构成的 n 元关系。也即选择满足条件的关系（记录）。
- 投影：投影 $P_{i_1,i_2,\cdots,i_m}$ 是将 n 元组 $a_1, a_2, \cdots, a_n$ 映射到 m 元组，$m \leq n$. 也即将 n 元组中位置为 $i_1, i_2, \cdots, i_m$ 的元素保留，其它删去。

  例如，数学老师只选取某个班级期末成绩单中的姓名、学号、数学成绩，而删去其它科目的成绩。
- 连接：设 R 是 m 元关系，S 是 n 元关系，$p \leq m, p \leq n$，则连接 $J_p(R,S)$ 是 $m+n-p$ 元关系，包含 $(a_1, a_2, \cdots, a_{m-p}, c_1, c_2, \cdots, c_p, b_1, b_2, \cdots, b_{n-p})$，其中，$m$

元组 $(a_1, a_2, \cdots, a_{m-p}, c_1, c_2, \cdots, c_p)$ 属于 R，n 元组 $(c_1, c_2, \cdots, c_p, b_1, b_2, \cdots, b_{n-p})$ 属于 S

例如，关系 R 为由 (课程, 课程号) 构成，关系 S 由 (课程号，教室号，时间) 构成。则连接 $J_1(R, S)$ 是由 (课程，课程号，教室号，时间) 构成，课程号为重叠的部分。

8.3 闭包

设 R 是集合 A 上的关系，R 可能具有或不具有某种性质 P。若存在关系 S 包含 R 具有某种性质 P，且 S 是最小的（也即 S 是所有的包含 R 具有某种性质 P 的关系的子集），则称 S 为 R 关于 P 的闭包。

例如集合 $A = \{1, 2, 3\}$，关系 $R = \{(1, 1), (2, 1), (3, 2)\}$ 不是自反的，我们增加两个元素 $(2, 2), (3, 3)$，构建关系 S 为自反的，称 S 是 R 的自反闭包。还有传递闭包、对称闭包。

传递闭包

设 R 是集合 A 上的关系，连通性关系 R^* 由对 (a, b) 构成，使得在 R 的有向图中存在一条长度至少为 1 的路径。也即，在关系 R 中不管中间经过多少元素，只要能从元素 a 传递到元素 b，则对 (a, b) 就会出现在 R^* 中。

- 设 R 是集合 A 上的关系。集合 A 的有向图中，a, b 之间存在长度为 n 的路径，等价于 $(a, b) \in R^n$。因此，连通性关系 $R^* = \bigcup_{i=1}^{n} R^i$.
- 关系 R 的传递闭包等于连通性关系 R^*.
- 设 R 是集合 A 上的关系，集合 A 有 n 个元素，若存在一条从 a 到 b 的长度至少为 1 的路径，则存在一条长度不超过 n 的这种路径。若 $a \neq b$，则则存在一条长度不超过 $n - 1$ 的这种路径。
- 设 $\mathbf{M_R}$ 是 n 元素集合上的关系 R 的 0-1 矩阵，则传递闭包 R^* 的 0-1 矩阵为

$$\mathbf{M_{R^*}} = \mathbf{M_R} \vee \mathbf{M_{R^2}} \vee \cdots \vee \mathbf{M_{R^n}}$$

沃舍尔算法

沃舍尔算法用来计算传递闭包，也称为罗伊-沃舍尔算法。
理解沃舍尔算法需要以下概念：

- 设 v_i 为集合的元素，$v_1, v_2, \cdots, v_n$ 为 n 个元素的任意排列。
- 内点：设路径 $a, x_1, x_2, \cdots, x_k, b$，其中 $x_1, x_2, \cdots, x_k$ 为 k 个内点。

设 R 为 n 元素集合上的关系，取集合中所有元素的一个排列，记为 $v_1, v_2, \cdots, v_n$. 沃舍尔算法需要构建一系列 n 阶 0-1 矩阵：$\mathbf{W_0}, \mathbf{W_1}, \cdots, \mathbf{W_n}$，其中，$\mathbf{W_k}= [w_{ij}^{[k]}]$. 当元素 v_i 到元素 v_j 之间存在一条路径，且该路径经过的所有内点为前 k 个元素 $v_1, v_2, \cdots v_k$ 的子集时，$w_{ij}^{[k]} = 1$，否则，$w_{ij}^{[k]} = 0$.

当 $k = 0$ 时，从 v_i 到 v_j 之间没有内点，因此 $\mathbf{W_0} = \mathbf{M_R}$；当 $k = n$ 时，从 v_i 到 v_j 之间的内点可以是集合中除起点终点外的任意元素，因此 $\mathbf{W_n} = \mathbf{M_{R^*}}$。

在已知 W_{k-1} 时，从 v_i 到 v_j 有两种情况：$(1)w_{ij}^{[k-1]} = 1 (2)w_{ij}^{[k-1]} = 0$. 要求 W_k，需要在可选内点的序列 $v_1, v_2, \cdots, v_{k-1}$ 中增加点 v_k.

1. 若 $w_{ij}^{[k-1]} = 1$，则 $w_{ij}^{[k]} = 1$. 即已经有路径存在。
2. 若 $w_{ij}^{[k-1]} = 0$，此时若 $w_{ik}^{[k-1]} = w_{kj}^{[k-1]} = 1$，即增加的点 v_k 使得可以从 v_i 传递到 v_k，再从 v_k 传递到 v_j，实现从 v_i 传递到 v_j，则 $w_{ij}^{[k]} = 1$.

综上所述，可得

$$\omega_{ij}^{[k]} = \omega_{ij}^{[k-1]} \vee (\omega_{ik}^{[k-1]} \wedge \omega_{kj}^{[k-1]})$$

由于已知 $\mathbf{W_0}$，依次计算 $\mathbf{W_1}, \mathbf{W_2}, \cdots, \mathbf{W_n}$，即可得到传递闭包 R^* 的 0-1 矩阵 $\mathbf{M_{R^*}} = \mathbf{W_n}$.

8.4 等价关系

如果集合 A 上的关系是自反的、对称的和传递的，则这个关系称为等价关系。

如果集合中的两个元素 a, b 因为等价关系而相联系，则称他们是等价的。

设 R 是集合 A 上的等价关系，与集合 A 上的元素 a 有关系的所有元素的集合叫做 a 的等价类。记作 $[a]_R$. 换句话说，$[a]_R = \{s \mid (a, s) \in R\}$.（如果命题中只讨论一个关系，如只考虑关系 R，等价类的下标可以省略。）

- 设 R 是集合 A 上的等价关系，则以下命题等价：$(1)aRb \quad (2)[a] = [b] \quad (3)[a] \cap [b] \neq \emptyset$.

  简单的证明：

  1. $(1) \to (2)$：当 $(a, b) \in R$，假设存在一个元素 $c \in A$ 使得 $(a, c) \in R$(此元素可能为 a 自身)，由对称性 $(c, a) \in R$，则由传递性 $(c, b) \in R$，可证，凡是属于 $[a]$ 的元素皆属于 $[b]$，也即 $[a] \subset [b]$。同理可证 $[b] \subset [a]$，因此 $[a] = [b]$.
  2. $(2) \to (3)$：当 $[a] = [b]$ 时，由于 R 的自反性，等价类必不为空，故存在元素假设记为 $c \in [a], c \in [b]$，故 $[a] \cap [b] \neq \emptyset$.
  3. $(3) \to (1)$：当 $[a] \cap [b] \neq \emptyset$，即存在元素假设记为 $c \in [a], c \in [b]$，也即 $(a, c) \in R, (b, c) \in R$。由对称性和传递性，$(a, b) \in R$.

  通过上述证明的组合，可得上述命题是等价的。

- 设 R 是集合 S 上的等价关系，那么 R 的等价类构成 S 的划分。相反，若给定集合 S 的一个划分 $\{A_i \mid i \in I\}$，则存在一个等价关系 R，其等价类为 $A_i (i \in I)$.

  R 上的等价类之间有两种情况 $(1)[a] \cap [b] = \emptyset (2)[a] \cap [b] \neq \emptyset$. 当 $[a] \cap [b] \neq \emptyset$ 时，由上述命题的等价关系 $[a] = [b]$，也即，R 上的等价类要么相同，要么没有交集。因此，R 上的等价类将 S 中的划分成不同的部分，在同一部分中的元素的等价类相同，为该部分；不同部分之间的元素的等价类没有交集。

  当给定一个划分时，每一个子集是一个等价类，通过等价类形成等价关系。

8.5 偏序

如果集合 S 上的关系 R 是自反的、反对称的和传递的，则这个关系称为偏序。(S, R) 称为偏序集。

- 偏序集 $(S, \preccurlyeq)$ 中，如果 $a \preccurlyeq b$ 或 $b \preccurlyeq a$，则称元素 a, b 是可比的；否则，称为不可比的。
- 偏序集 $(S, \preccurlyeq)$ 中，如果 S 中的每对元素都可比，则 S 称为全序集或线序集，也称为链。此时 $\preccurlyeq$ 叫做全序或线序。
- 偏序集 $(S, \preccurlyeq)$ 中，如果 $\preccurlyeq$ 全序，且 S 中的每个非空子集都有最小元素，称它为良序集。

良序归纳定理：设集合 S 是一个良序集。如果对于所有的 $y \in S$，当对于所有的 $x \in S, x \prec y$，$P(x)$ 均成立时，$P(y)$ 成立，则对于所有的 $x \in S$，$P(x)$ 恒成立。

其它的一些概念：

- 字典顺序：两个序列中第一个元素小的在前，如果相等，则第二个元素小的在前，以此类推。
- 哈塞图：一个偏序集中的集合的有向图，去除表示自反的环（元素指向自身），去除可以用更短的路径组合出的表示传递的路径，将起点放在最下、终点在最上，最后化简为一个包含足够信息的图，称为哈塞图。
- 极大值和极小值
- 上界和下界：对于偏序集 $(S, \preccurlyeq)$ 的子集 A，若存在元素 $u \in S$ 使得 A 中的所有元素 $a_i \preccurlyeq u$，则称 u 为 A 的一个上界。下界的定义类似。
- 最小上界和最大下界。在子集 A 的所有上界中，如果上界 x 小于其它任何的上界，则称 x 为 A 的最小上界。同理，在子集 A 的所有下界中，如果下界 x 大于其它任何的下界，则称 x 为 A 的最大下界。分别记作 $\mathrm{glb}(A)$ 和 $\mathrm{lub}(A)$.
- 若一个偏序集中的每一对元素都有最小上界和最大下界，则称此偏序集为格。

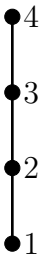


图 1: $(\{1,2,3,4\},\leq)$ 的哈塞图

拓扑排序

如果只要 aRb 就有 $a \preccurlyeq b$，则称一个全序 $\preccurlyeq$ 和一个偏序 R 是相容的。从一个偏序构造一个相容的全序的过程称为拓扑排序。

每个有穷非空偏序集都有极小元素。

在又穷的偏序集 $(A,\preccurlyeq)$ 中，由上述结论，选择一个最小元素 a_1；接下来对于偏序集 $(A-\{a_1\},\preccurlyeq)$，若其非空，再从中选择一个最下元素 a_2；接下来对于偏序集 $(A-\{a_1,a_2\},\preccurlyeq)$，若其非空，再从中选择一个最小元素 a_3；…因为 A 是有穷集合，此过程必能结束。得到最终序列（全序）

$$a_1, a_2, \cdots, a_n$$

9 图

9.1 图的术语与表示

图 (V,E) 由非空顶点集 V 和边集 E 组成，分为无向图和有向图。

没有环和多重边的图为简单图。

基本概念

在无向图中，由一条边连接或关联的两个顶点 u,v 称为邻接，可以记作 $\{u,v\}$，u,v 称为该边的端点。

在有向图中，由一条边连接 u,v 时，称为 u 邻接到 v，可以记作 (u,v). 此时，u 为起点，v 为终点。

顶点的度：

- 无向图中顶点的度指与该顶点关联的边的数目，特殊的，顶点上的环为该顶点贡献双倍的度。记作 $\deg(v)$.
- 有向图中顶点的入度是以该顶点为终点的边的数目，记作 $\deg^-(v)$. 顶点的出度是以改顶点为起点的边的数目，记作 $\deg^+(v)$.

性质

- 握手定理：设 $G=(V,E)$ 是有 n 条边的无向图，则 $2n=\sum_{v\in V}\deg(v)$.
- 无向图有偶数个奇数度的顶点。
- 设 $G=(V,E)$ 是有向图，则 $\sum_{v\in V}\deg^-(v)=\sum_{v\in V}\deg^+(v)=|E|$.
- 子图和并图

一些特殊的简单图

- 完全图 K_n：每对不同顶点之间都恰有一条边的简单图。
- 圈图 C_n：由顶点 $v_1,v_2,\cdots,v_n$ 和边 $\{v_1,v_2\},\{v_2,v_3\},\cdots,\{v_{n-1},v_n\},\{v_n,v_1\}$ 组成的简单图。
- 轮图 W_n：轮图 W_n 是在圈图 C_n 的基础上增加一个顶点，该顶点与圈图中的所有顶点相连。（注意：W_n 有 $n+1$ 个顶点。）
- n 立方图 Q_n：用 2^n 个顶点表示 n 位二进制位串，当两个位串只相差一位，其对应的顶点相连。

偶图

如果可以把简单图 G 的顶点分成两个不相交的非空集合 V_1,V_2，使得 G 的所有边的一个端点来自 V_1，另一个端点比来自 V_2，满足这样条件的图 G 称为偶图。此时，称 (V_1,V_2) 为 G 的二部划分。

如果一个简单图的每个顶点可以染成两种颜色中的一种，且相邻接的顶点不被染成同一种颜色，等价于该图为一个偶图。

图的表示

- 对于简单图，可以使用邻接矩阵。需要将图的顶点排序，作为矩阵中元素的行号和列号，无向图的邻接矩阵中使用 0-1 值表示两个点之间是否有边，无向图的邻接矩阵是一个对称阵；有向图的邻接矩阵中使用 0-1 值表示从行号到列号是否有边，有向图的邻接可以是非对称阵。对于非简单图，只需将矩阵中的 0-1 值换成边数即可。
- 对于稀疏的图，可以使用邻接表，节省空间。邻接表中的表项，id 为每个顶点，内容为与改顶点直接相连接的其它顶点。

- 无向图还可以使用关联矩阵，将图的顶点排序作为矩阵的行号，边排列作为矩阵的列号，矩阵中的元素值表示其行号对应顶点和列号对应边之间是否相关联，或者多重边时表示重数。

同构

设 $G_1 = (V_1, E_1)$ 和 $G_2 = (V_2, E_2)$ 是简单图，若存在一对一的和映上的从 V_1 到 V_2 的函数 F，且 f 对于 V_1 里所有的 a 和 b 来说，a, b 在 G 里相邻当且仅当 $f(a), f(b)$ 在 G_2 里相邻，就称 G_1, G_2 是同构的，这样的函数 f 称为同构。

判断两个图是否同构很难，但是判断两个图不同构相对简单。若两个图有不一样的某条性质，如不同数量的边或顶点、不同数量的相同度的边、不同数量的相同长度的回路等，则两个图不同构。

9.2 连通性

通路是边的序列。若通路或回路不重复第包含相同的边，则它是简单的。

- 若无向图每一对不同的顶点之间都有通路，则该图称为连通的。在连通无向图的每对顶点之间都存在简单通路。

- 若有向图每一对不同的顶点，如 (a, b)，有从 a 到 b 和从 b 到 a 的通路，则该图是强连通的。若在有向图的底图中，任何两个顶点之间都有通路，则该有向图是若连通的。

- 底图：把有向图中的每条有向边改为无向边，所得的无向图是该有向图的底图。

- 设 G 是带有相对于顶点顺序 $v_1, v_2, \cdots, v_n$ 的邻接矩阵 $\mathbf{A}$ 的图（允许带有无向或者有向边、多重边和环），从 v_i 到 v_j 的长度为 r 的不同通路的数目等于 $\mathbf{A^r}$ 的第 (i, j) 项，其中 r 是正整数。

欧拉通路和哈密顿通路

- 欧拉回路、欧拉通路：图 G 的欧拉回路是包含 G 所有边的简单回路，图 G 的欧拉回路是包含 G 所有边的简单通路。

  连通多重图具有欧拉回路当且仅当它的每个顶点都有偶数度。

  连通多重图具有欧拉通路但是没有欧拉回路当且仅当它恰有 2 个奇数度顶点。

- 哈密顿通路、哈密顿回路：设图 G 里，经过所有顶点但是不重复经过同一个顶点的通路，称为哈密顿通路。在哈密顿通路的基础上，增加由终点到起点的边构成的回路，称为哈密顿回路。

不能找到一个哈密顿回路的等价描述，但是有一些判断图有哈密顿回路的充分条件：

- 狄拉克定理：若 G 是带有 n 个顶点的连通简单图，$n \geq 3$，并且 G 中每个顶点的度都至少为 $n/2$，则 G 有哈密顿回路。
- 奥尔定理：若 G 是带有 n 个顶点的连通简单图，$n \geq 3$，并且对于 G 中每一对不相邻的顶点 u, v 来说，都有 $\deg(u) + \deg(v) \geq n$，则 G 有哈密顿回路。

国际象棋中马在 $m \times n$ 的棋盘上的周游等价于求表示马在该棋盘上合法移动的图的哈密顿通路。

带权图的最小通路

给每条边赋上一个数的图称为带权图。

求带权图两个顶点之间的最短通路，比如旅行商问题。有不同的算法，如迪克斯特拉算法。以下图为例：

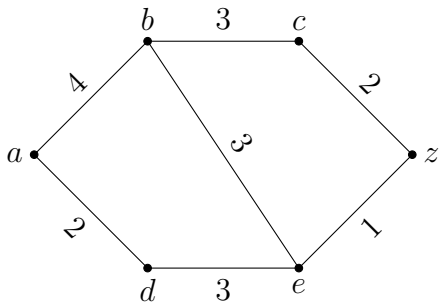


图 2: 带权的简单图

$\{a, b\} > \{a, d\}, min = \{a, d\}$

$\{a, b\} < \{a, d, e\}, min = \{a, b\}$

$\{a, b, c\} = \{a, b, e\} > \{a, d, e\}, min = \{a, d, e\}$

$\{a, b, c\} = \{a, b, e\} > \{a, d, e, z\}, min = \{a, d, e, z\}$

end.

迪克斯特拉算法求出的连通简单无向带权图里两个顶点之间的最短通路的长度。需要 $O(n^2)$ 次运算。

9.3 可平面的图

若可以在平面里画出一个图且边没有任何交叉，则这个图是可平面的，这种画法称为这个图的平面表示。

- 欧拉公式：设 G 是带 e 条边和 v 个顶点的连通可平面简单图，设 r 是 G 的可平面表示里的面数，则 $r = e - v + 2$.
- 设 G 是带 e 条边和 v 个顶点的连通可平面简单图，$v \geq 3$，则 $e \leq 3v - 6$.
- 设 G 是连通可平面简单图，则 G 有度数不超过 5 的点。
- 设 G 是带 e 条边和 v 个顶点的连通可平面简单图，$v \geq 3$，且没有长度为 3 的回路，则 $e \leq 2v - 4$.

若一个图是可平面的，则通过删除一条边 $\{u, v\}$ 并且添加新顶点 w 和两条边 $\{u, w\}, \{w, v\}$，所获得的图也是可平面的，这样的操作称为初等细分。若可以从相同的图通过一系列初等细分来获得两个图，则它们称为同胚的。

库拉图斯基定理：一个图是非可平面的当且仅当它包含一个同胚于 $K_{3,3}$ 或 K_5 的子图，因为 $K_{3,3}$ 和 K_5 是不可平面的。

9.4 图着色

简单图的着色是对该图的每个顶点指定一种颜色，使得没有两个相邻的顶点颜色相同。图的色数是着色这个图所需要的最少颜色数。

四色定理：平面图的色数不超过 4.

- 完全图 K_n 的色数为 n.
- 完全偶图 $K_{m,n}$ 的色数为 2.
- 圈图 C_n 的色数，当 n 为偶数，色数为 2；当 n 为大于 1 的奇数，色数为 3.

拉姆塞定理：拉姆塞证明任何 6 个人的聚会，其中总会有 3 个人相互认识，或 3 个人相互不认识，其中 6 称为拉姆塞数，记作 $r(3, 3) = 6$。拉姆塞问题是着色问题，可以表述为具有 n 个顶点的完全图，使用红蓝两种颜色着色，则可以找到一个具有 s 个顶点的红色完全图，或者一个具有 t 个顶点的蓝色完全图。

- $r(s, t) = r(t, s)$
- $r(s, 0) = 0, r(s, 1) = 1, r(s, 2) = s$
- $r(3, 3) = 6, r(4, 4) = 9$，但是 $r(5, 5)$ 及后面的就很难求出。
- 若 $k \geq 2$，则 $r(k, k) \geq 2^{k/2}$

10 树

树是没有简单回路的连通无向图。一个无向图是树当且仅当在它的每对顶点之间都存在唯一简单通路。

- 根树是指定一个顶点作为根并且每条边的方向都离开根的树。
- 有子女的顶点称为内点，根是内点。没有子女的顶点称为树叶。
- 若根树的每个内点都有不超过 m 个子女，则称它为 m 元树。若该树的每个内点都恰好有 m 个子女，则称它为正则 m 元树。$m = 2$ 时的正则 m 元树称为二叉树。
- 在根树里顶点 v 的层是从根到该顶点的唯一通路的长度，根的曾定义为 0。根树的高度就是顶点层数的最大值。
- 有序根树是把每个内点的子女们都排序的根树。

常用结论

- 带有 n 个顶点的树含有 $n-1$ 条边。因为每一个不是根的顶点，都有唯一的一条向上的边，且这些边构成了树中所有的边。
- 带有 i 个内点的正则 m 元树含有 $n = mi + 1$ 个顶点。
- 一个正则 m 元树的顶点 n、内点 i、树叶 l 之间满足：

$$i + l = n$$

$$n = mi + 1$$

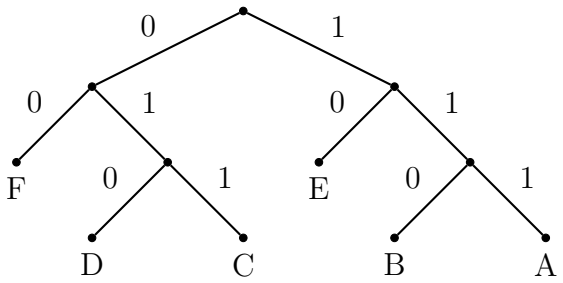
$$l = n - i = mi + 1 - i = (m-1)i + 1$$

- 在高度为 h 的 m 元树里至多有 m^k 个树叶。
- 若一个高度为 h 的 m 元树有 l 个树叶，则 $h \geq \lceil \log_m l \rceil$。若这个 m 元树是正则的和平衡的，则 $h = \lceil \log_m l \rceil$.

树的应用

- 二叉搜索树：基于一系列比较确定一个元素的位置（二分搜索），若树是平衡的，则确定一个项的位置或者添加一个项所需的比较次数不超过 $\lceil \log(n+1) \rceil$.
- 决策树：通过一步一步的决策构建问题的解，从根到树叶的通路都是问题的可能解。可以使用决策树来确定排序算法的最坏情形复杂度等。

- 前缀码：使用不同长度的位串来编码字母，较短的位串用于编码频繁的字母，较长的位串用于编码不经常出现的字母。如赫夫曼编码：



- 博弈树：博弈树通常用于描述二人轮流的游戏，每个树叶表示游戏结局的一种。

  1. 每个树叶为游戏结束时第一个玩家的分数；
  2. 偶数层顶点的值选择子女中的最大值，奇数层的值顶点选择子女中的最小值。因为第二个玩家的回合总是想让第一个玩家的分数降低。这种策略称为最小最大策略。若两个玩家都严格遵循最小最大策略，则可以计算哪位玩家获胜。

10.1 遍历

通用地址系统：用整数 0 标记根，然后用 $1, 2, 3, \cdots, k$ 从左向右标记它的 k 个子女；对于一个已经标记为 A 的内点，从左向右依次标记它的 k 个子女为 $A.1, A.2, \cdots, A.k$.

系统地访问有序根树的每个顶点的过程称为遍历算法。三个最常用的遍历算法是：前序遍历、中序遍历、后序遍历。

- 前序遍历：设 T 是带根 r 的有序根树。若只包含 r，则 r 是 T 的前序遍历。否则，假定 $T_1, T_2, \cdots, T_k$ 是 r 的从左向右的子树，前序遍历首先访问 r，再依次以前序来遍历$T_1, T_2, \cdots, T_k$.

- 中序遍历：设 T 是带根 r 的有序根树。若只包含 r，则 r 是 T 的中序遍历。否则，假定 $T_1, T_2, \cdots, T_k$ 是 r 的从左向右的子树，中序遍历首先以中序来遍历T_1，再访问 r，最后依次以中序来遍历$T_2, \cdots, T_k$.

- 后序遍历：设 T 是带根 r 的有序根树。若只包含 r，则 r 是 T 的后序遍历。否则，假定 $T_1, T_2, \cdots, T_k$ 是 r 的从左向右的子树，后序遍历首先依次以后序来遍历$T_1, T_2, \cdots, T_k$，再访问 r.

表达式的前缀、中缀和后缀记法

对表达式的二叉树的前序、中序和后序遍历形成其前缀、中缀和后缀记法。前缀形式也称作波兰记法。

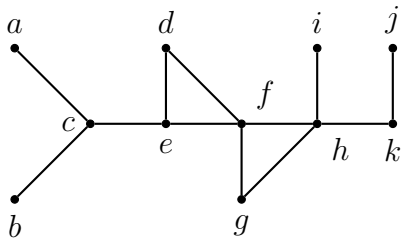
10.2 生成树和最小生成树

生成树

设 G 是简单图，G 的生成树是包含 G 的每个顶点的 G 的子图。再 G 的基础上删去一些边来形成树。

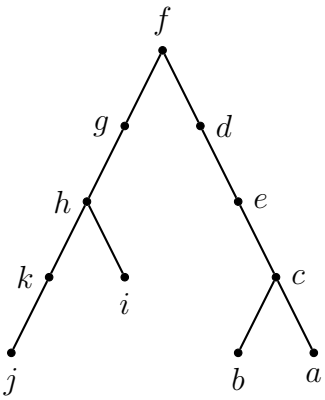
简单图是连通的当且仅当它具有生成树。

构建生成树的算法分为深度优先搜索算法和宽度优先搜索算法。以下图为例：



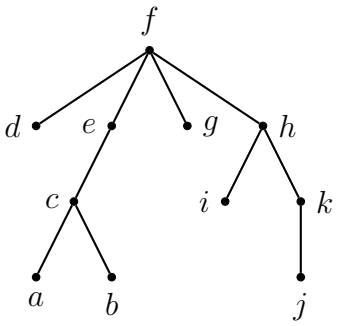
深度优先搜索

以 f 为起点构建一条到一个树叶的通路，顶点不重复，如选择 $f-g-h-k-j$ 做为起始的树。从最后的顶点回溯，不存在以 k 开始且含有不在树上的顶点的通路；然后回溯到 h，存在以 h 开始且含有不在树上的顶点的通路 $h-i$，将它添加到树上；依次向上回溯，可以构建出如下的树：



宽度优先搜索

选取 f 为根，将与 f 相邻的所有的顶点作为 f 的子女，添加到 1 层上。然后将于 1 层上的顶点相邻且不在树上的顶点，作为对于的顶点的子女，添加到 2 层上。依次类推，可以构建出如下的树：



最小生成树

连通带权图里的最小生成树是具有最小可能的边的权之和的生成树。
构建最小生成树可以使用普林算法和克鲁斯卡尔算法。

- 普林算法：首先选择带最小权的边，把它放进生成树里。相继地向树里添加与已在树里的顶点相关联的、并且不与已在树里的边形成简单回路的权最小的边。当已经添加了 $n-1$ 条边时停止。
- 克鲁斯卡尔算法：首先选择图中权最小的一条边，把它放进树里，相继地添加不与已在树里的边形成简单回路的权最小的边。当已经添加了 $n-1$ 条边时停止。

11 布尔代数

11.1 布尔代数的概念与表示

- 3 种常用的布尔运算：补 $\overline{x}$、和 $x+y$、积 $x\cdot y$ 对应 NOT、OR、AND.
- 布尔函数：$y=f(x_1,x_2,\cdots,x_n)$，其中 x 称为布尔变元，$x_1,x_2,\cdots,x_n,y\in\{0,1\}$.
- 以 X 表示 $x_1,x_2,\cdots,x_n$，有以下结论：
  - $\overline{F}(X)=\overline{F(X)}$
  - $(F+G)(X)=F(X)+G(X)$
  - $(FG)(X)=F(X)G(X)$
- 布尔恒等式：
  - $\overline{\overline{x}}=x$
  - $x+x=x, x\cdot x=x$
  - $x+0=x, x\cdot 1=x$
  - $x+1=1, x\cdot 0=0$

- $x+y=y+x, xy=yx$
- $x+(y+z)=(x+y)+z, x(yz)=(xy)z$
- $x+yz=(x+y)(x+z), x(y+z)=xy+xz$
- $\overline{(xy)}=\overline{x}+\overline{y}, \overline{(x+y)}=\overline{xy}$
- $x+xy=x, x(x+y)=x$
- $x+\overline{x}=1, x\overline{x}=0$

恒等式中经常有两个式子同时出现，这种关系可以称为对偶。可以交换积与和，交换 0 与 1，来获得一个式子相对偶的式子。

- 文字：布尔变元或者它的补称为文字。

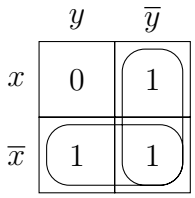
  小项：n 个文字的积。

  布尔表达式可以化简成多个小项的和，称为积之和展开式或析取范式。

电路最小化

将一个布尔表达式化简成最简的积之和展开式。这时实现这样的逻辑所需的电路元件最少。

- 卡诺图：适用于布尔变元较少的情况。以两个布尔变元为例：



  上图中圆角矩形横跨了 $\overline{x}$ 的一行，此时表明 $\overline{x}$ 必会发生，可以将圆角矩形中两个格子代表的元素合成一个 $\overline{x}$。同理，另一个圆角矩形将两个格子合成 $\overline{y}$，因此，这个图表示的是 $x\overline{y}+\overline{x}y+\overline{x}\cdot\overline{y}=\overline{x}+\overline{y}$.

- 奎因-莫可拉斯基方法当布尔变元较多时，卡诺图的方法就不方便。可以使用奎因-莫克拉斯基方法。首先将 n 个变元构成的小项，用 n 位二进制位串表示，其中 x 用 1 表示，$\overline{x}$ 用 0 表示。然后根据串中 1 的个数将这些串分组，仅相差一位二进制的串可以合并，合并后的串中不同的那位二进制用'-' 表示。只要能够合并就继续合并，直到不能合并为止。将合并的结果还原回小项，其中的'-' 表示没有这一变元。

12 计算模型

12.1 有限状态机

- 有限状态机：设 $M=(S,I,O,f,g,s_0)$，其中 S 表示有限状态的集合，I 表示输入字母表，每个字母代表一种输入，O 表示输出字母表，每个字母代表一种输出。转移函数 f 输入一种状态和输入字母，就会产生一种状态。输出函数 g 输入一种状态和输入字母，就会产生一种输出字母。s_0 表示初始状态。

  有限状态机可以使用状态表或者带有标号边的有向图来表示。

  设有限状态机 M，当存在一个集合 $L\subset I$ 且 L 中的所有元素作为 M 的输入时，M 的输出均为 1，称 M 能够识别（或接受）L.

  有限状态机可以分为米利机和摩尔机。米利机的输出与初始状态和输出都有关系，而摩尔机的输出只与初始状态有关系。

- 有限状态自动机：是一种没有输出的有限状态机。设 $M=(S,I,f,s_0,F)$，其中 F 为终结状态。

  转移函数扩展：若 $f(s,xy)=f(f(s,x),y)$，则称该状态机可以转移函数扩展。

- 非确定型有限状态自动机：设 $M=(S,I,f,s_0,F)$，与有限状态自动机不同的是，转移函数给出的输出状态可能是多个状态中的一个。

- 更强大的机器：下推自动机、线性有界自动机、图灵机

12.2 语言处理

- 词汇表 V^*.

- 词汇表在一定的文法下产生的所有符串的集合记作 $L(G)$.

- 克莱因闭包：设 $A\subset V^*$，则 A 的克莱因闭包是 A 中的任意多个串的组合而形成发集合，记作 A^*。

- 如果语言 L 可以由一个非确定型有限状态自动机识别，则它必定可以由一个确定型有限状态自动机识别。

- 正则集合：从空集、空串、单字符串开始，以任意顺序连接，并和克莱因闭包运算所得的。

- 克莱因定理：一个集合是正则的，当且仅当它可以由一个有限状态机识别。

- 一个集合可以由正则文法形成，当且仅当它是一个正则集合。

12.3 计算复杂度、可计算性、可判定性

- 判定问题：是与不是的问题。

  可判定性：当存在一个算法来判断判定问题的某个解是否正确，则称这个问题是可判定的。

- 停机问题：停机问题是不解的判定问题，当给定图灵机 T 的编码及输入串 x，没有图灵机能够判定该图灵机 T 最终是否会停机。

- 可计算性：如果一个函数可以被图灵机计算，则称这个函数是可计算的。

- P 类问题：如果一个判定问题可以用确定型图灵机在多项式时间内求解，则这个问题是 P 类问题，即确定型多项式时间问题。

  NP 类问题：如歌一个判定问题可以用非确定型图灵机在多形式时间内求解，则这个问题是 NP 类问题，即非确定型多项式时间问题。

  属于 P 类问题的是易处理的，不属于 P 类问题的是不易处理的。P$\subset$NP.