

基础数论定理

2018年10月25日 11:20

【裴蜀定理】

Bézout's identity — Let a and b be integers with greatest common divisor d . Then, there exist integers x and y such that $ax + by = d$. More generally, the integers of the form $ax + by$ are exactly the multiples of d .

When one pair of Bézout coefficients (x, y) has been computed (e.g., using [extended Euclidean algorithm](#)), all pairs can be represented in the form

$$\left(x + k \frac{b}{\gcd(a, b)}, y - k \frac{a}{\gcd(a, b)} \right),$$

where k is an arbitrary integer and the fractions simplify to integers.

For three or more integers [\[edit\]](#)

Bézout's identity can be extended to more than two integers: if

$$\gcd(a_1, a_2, \dots, a_n) = d$$

then there are integers x_1, x_2, \dots, x_n such that

$$d = a_1 x_1 + a_2 x_2 + \dots + a_n x_n$$

has the following properties:

- d is the smallest positive integer of this form
- every number of this form is a multiple of d

【欧拉数论定理】

if n and a are ***coprime*** positive integers, then

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

降幂公式:

$$A^K \equiv A^{K \% \phi(m) + \phi(m)} \pmod{m} \quad K > \phi(m)$$

【卢卡斯定理】

For non-negative integers m and n and a prime p , the following congruence relation holds:

$$\binom{m}{n} \equiv \prod_{i=0}^k \binom{m_i}{n_i} \pmod{p},$$

where

$$m = m_k p^k + m_{k-1} p^{k-1} + \cdots + m_1 p + m_0,$$

and

$$n = n_k p^k + n_{k-1} p^{k-1} + \cdots + n_1 p + n_0$$

are the base p expansions of m and n respectively. This uses the convention that $\binom{m}{n} = 0$ if $m < n$.

【中国剩余定理】

设正整数 m_1, m_2, \dots, m_k 两两互素, 则同余方程组

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

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

$$x \equiv a_3 \pmod{m_3}$$

·

·

·

$$x \equiv a_k \pmod{m_k}$$

有整数解。并且在模 $M = m_1 \cdot m_2 \cdot \dots \cdot m_k$ 下的解是唯一的, 解为

$$x \equiv (a_1 M_1 M_1^{-1} + a_2 M_2 M_2^{-1} + \dots + a_k M_k M_k^{-1}) \pmod{M}$$

其中 $M_i = M/m_i$, 而 M_i^{-1} 为 M_i 模 m_i 的逆元。

中国剩余定理扩展——求解模数不互质情况下的线性方程组：

普通的中国剩余定理要求所有的 m_i 互素，那么如果不互素呢，怎么求解同余方程组？

这种情况就采用两两合并的思想，假设要合并如下两个方程：

$$x = a_1 + m_1 x_1$$

$$x = a_2 + m_2 x_2$$

那么得到：

$$a_1 + m_1 x_1 = a_2 + m_2 x_2 \Rightarrow m_1 x_1 + m_2 x_2 = a_2 - a_1$$

我们需要求出一个最小的 x 使它满足：

$$x = a_1 + m_1 x_1 = a_2 + m_2 x_2$$

那么 x_1 和 x_2 就要尽可能的小，于是我们用扩展欧几里得算法求出 x_1 的最小正整数解，将它代回 $a_1 + m_1 x_1$ ，得到 x 的一个特解 x' ，当然也是最小正整数解。

所以 x 的通解一定是 x' 加上 $\text{lcm}(m_1, m_2) * k$ ，这样才能保证 x 模 m_1 和 m_2 的余数是 a_1 和 a_2 。由此，我们把这个 x' 当做新的方程的余数，把 $\text{lcm}(m_1, m_2)$ 当做新的方程的模数。（这一段是关键）

合并完成：

$$x \equiv x' \pmod{\text{lcm}(m_1, m_2)}$$

【威尔逊定理】

p 可整除 $(p-1)! + 1$ 是 p 为质数的充要条件

完全剩余系和简化剩余系

2018年10月25日 12:00

【完全剩余系】

在模 n 的剩余类中各取一个元素，则这 n 个数就构成了模 n 的一个完全剩余系。

性质一

对于 n 个整数，其构成模 n 的完系等价于其关于模 n 两两不同余；

性质二

若 $a_i (1 \leq i \leq n)$ 构成模 n 的完系， $k, m \in \mathbb{Z}$, $(m, n)=1$, 则

$$k + ma_i \ (1 \leq i \leq n)$$

也构成模 n 的完系；

性质三

若 $a_i (1 \leq i \leq n)$ 构成模 n 的完系，则

$$\sum_{i=1}^n a_i = \frac{n(n+1)}{2} = \begin{cases} \frac{n}{2} \pmod{n} \\ 0 \pmod{n} \end{cases}$$

【简化剩余系】

Any subset R of the integers is called a **reduced residue system** modulo n if:

1. $\gcd(r, n) = 1$ for each r contained in R ;
2. R contains $\varphi(n)$ elements;
3. no two elements of R are congruent modulo n .^{[1][2]}

Here φ denotes [Euler's totient function](#).

- If $\{r_1, r_2, \dots, r_{\varphi(n)}\}$ is a reduced residue system with $n > 2$, then $\sum r_i \equiv 0 \pmod{n}$.
- Every number in a reduced residue system mod n is a generator for the additive group of integers modulo n .

离散对数和原根

2018年10月25日 13:36

【原根】

原根是一种数学符号，设 m 是正整数， a 是整数，若 a 模 m 的阶等于 $\phi(m)$ ，则称 a 为模 m 的一个原根。

原根具有以下性质：

(1) 可以证明，如果正整数 $(a, m) = 1$ 和正整数 d 满足 $a^d \equiv 1 \pmod{m}$ ，则 d 整除 $\phi(m)$ 。因此 $\text{Ord}_m(a)$ 整除 $\phi(m)$ 。在例子中，当 $a=3$ 时，我们仅需要验证3的1、2、3和6次方模7的余数即可。

(2) 记 $\delta = \text{Ord}_m(a)$ ，则 $a^1, \dots, a^{(\delta-1)}$ 模 m 两两不同余。因此当 a 是模 m 的原根时， $a^0, a^1, \dots, a^{(\delta-1)}$ 构成模 m 的简化剩余系。

(3) 模 m 有原根的充要条件是 $m = 1, 2, 4, p, 2p, p^n$ ，其中 p 是奇质数， n 是任意正整数。

(4) 对正整数 $(a, m) = 1$ ，如果 a 是模 m 的原根，那么 a 是整数模 n 乘法群（即加法群 $\mathbb{Z}/m\mathbb{Z}$ 的可逆元，也就是所有与 m 互素的正整数构成的等价类构成的乘法群） \mathbb{Z}_n 的一个生成元。由于 \mathbb{Z}_n 有 $\phi(m)$ 个元素，而它的生成元的个数就是它的可逆元个数，即 $\phi(\phi(m))$ 个，因此当模 m 有原根时，它有 $\phi(\phi(m))$ 个原根。 [2]

原根存在的条件有以下几个：

定理一：设 p 是奇素数，则模 p 的原根存在； [3]

定理二：设 g 是模 p 的原根，则 g 或者 $g+p$ 是模 p^2 的原根；

定理三：设 p 是奇素数，则对任意 α ，模 p^α 的原根存在；

定理四：设 $\alpha \geq 1$ ，则 g 是模 p^α 的一个原根，则 g 与 $g+p^\alpha$ 中的奇数是模 $2p^\alpha$ 的一个原根。

模 m 有原根的充要条件： $m = 2, 4, p^a, 2p^a$ ，其中 p 是奇素数。

求模素数 p 原根的方法：对 $p-1$ 素因子分解，即 $p-1 = p_1^{a_1} p_2^{a_2} \dots p_k^{a_k}$ 是 $p-1$ 的标准分解式，若恒有

$$g^{\frac{p-1}{p_i}} \not\equiv 1 \pmod{p}$$

成立，则 g 就是 p 的原根。（对于合数求原根，只需把 $p-1$ 换成 $\phi(p)$ 即可）

【离散对数】

Input: A cyclic group G of order n , having a generator α and an element β .

Output: A value x satisfying $\alpha^x = \beta$.

1. $m \leftarrow \text{Ceiling}(\sqrt{n})$
2. For all j where $0 \leq j < m$:
 1. Compute α^j and store the pair (j, α^j) in a table. (See section "In practice")
3. Compute α^{-m} .
4. $\gamma \leftarrow \beta$. (set $\gamma = \beta$)
5. For all i where $0 \leq i < m$:
 1. Check to see if γ is the second component (α^j) of any pair in the table.
 2. If so, return $im + j$.
 3. If not, $\gamma \leftarrow \gamma \cdot \alpha^{-m}$.

```
#include <cmath>
#include <cstdint>
#include <unordered_map>
std::uint32_t pow_m(std::uint32_t base, std::uint32_t exp, std::uint32_t mod) {
    // modular exponentiation using the square-multiply-algorithm
}
/// Computes x such that g^x % mod == h
std::optional<std::uint32_t> babystep_giantstep(std::uint32_t g, std::uint32_t h,
std::uint32_t mod) {
    const auto m = static_cast<std::uint32_t>(std::ceil(std::sqrt(mod)));
    auto table = std::unordered_map<std::uint32_t, std::uint32_t>{};
    auto e = std::uint64_t{1}; // temporary values may be bigger than 32 bit
    for (auto i = std::uint32_t{0}; i < m; ++i) {
        table[static_cast<std::uint32_t>(e)] = i;
        e = (e * g) % mod;
    }
    const auto factor = pow_m(g, mod-m-1, mod);
    e = h;
    for (auto i = std::uint32_t{}; i < m; ++i) {
        if (auto it = table.find(static_cast<std::uint32_t>(e)); it !=
table.end()) {
            return {i*m + it->second};
        }
        e = (e * factor) % mod;
    }
    return std::nullopt;
}
```


数学大猜想和大定理

2018年10月25日 14:57

【费马大定理】

当整数 $n > 2$ 时，关于 x, y, z 的方程 $x^n + y^n = z^n$ 没有正整数解。

【费马平方和定理】

费马平方和定理的表述是：奇质数能表示为两个平方数之和的充分必要条件是该质数被4除余1。

【拉格朗日四平方和定理】

四平方和定理（Lagrange's four-square theorem）说明每个正整数均可表示为4个整数的平方和。它是费马多边形数定理和华林问题的特例。注意有些整数不可表示为3个整数的平方和，例如7。

【费马多边形数定理】

费马多边形数定理是一个定律，定义为每一个正整数都可以表示为最多 n 个 n 边形数的和。也就是说，每一个正整数一定可以表示为不超过三个的三角形数之和、不超过四个的平方数之和、不超过五个的五边形数之和，依此类推。

【华林问题】

华林问题是数论中的问题之一。1770年，爱德华·华林猜想，对于每个非1的正整数 k ，皆存在正整数 $g(k)$ ，使得每个正整数都可以表示为至多 $g(k)$ 个 k 次方数（即正整数的 k 次方）之和。

【空间分割定理】

n 个 $(k-1)$ 维空间最多能将一个 k 维空间分割成 $L(n, k)$ 个部分（这里说的空间皆为平直空间）。其中 $L(n, k)$ 满足以下性质：

1、定义域： $n \in \mathbb{N}, k \in \mathbb{N}^+$ 。

2、初始值： $L(0, k)=1, L(n, 1)=n+1$ 。

3、递推关系： $L(n, k)=L(n-1, k)+L(n-1, k-1)$ 。

$L(n, k)$ 有一个简洁的表达式，即：
$$L(n, k) = \sum_{m=0}^k \sum_{m=0}^k C(n, m)$$

以上为空间分割定理。 [2]

【勒让德定理】

在正数 $n!$ 的素因子标准分解式中，素数 p 的最高指数记作 $L_p(n!)$ ，则
$$L_p(n!) = \sum_{k \geq 1} \left\lfloor \frac{n}{p^k} \right\rfloor$$

【齐肯多夫定理】

任何正整数都可以表示成若干个不连续的斐波那契数（不包括第一个斐波那契数）之和。这种和式称为齐肯多夫表述法。

【棣莫弗定理】

设两个复数（用三角形表示） $z_1 = r_1(\cos\theta_1 + i\sin\theta_1)$, $z_2 = r_2(\cos\theta_2 + i\sin\theta_2)$ ，则：

$$z_1 z_2 = r_1 r_2 [\cos(\theta_1 + \theta_2) + i\sin(\theta_1 + \theta_2)]$$

【二次互反律】

二次互反律是经典数论中最出色的定理之一。二次互反律涉及到平方剩余的概念。设a,b是两个非零整数，我们定义雅克比符号 $\left(\frac{a}{b}\right)$ ：若存在整数x,使得 $x^2 \equiv a \pmod{b}$ ，那么就记 $\left(\frac{a}{b}\right) = 1$ ；否则就记 $\left(\frac{a}{b}\right) = -1$ 。在b是素数时这个符号也叫做勒让德符号。 [2]

高斯二次互反律：

设p和q为不同的奇素数，则 $\left(\frac{q}{p}\right) \cdot \left(\frac{p}{q}\right) = (-1)^{(p-1)/2 \cdot (q-1)/2}$

【15定理】

如果一个二次多项式可以通过变量取整数值而表示出1~15的值（更严格的结论是只要表示出1, 2, 3, 5, 6, 7, 10, 14, 15）的话（例如 $a^2 + b^2 + c^2 + d^2$ ），该二次多项式可以通过变量取整数值而表示出所有正整数。 [1]

【米迪定理】

米迪定理说明若有质数p、少于p的正整数a、大于1的正整数b和任意正整数n，

使得 a/p 在b进位制内的循环节长度是 $2n$ ，且将这个分数用循环小数写成 $0.\overline{a_1 a_2 a_3 \dots a_n a_{n+1} \dots a_{2n}}$ ，则有以下结论：

$$\begin{aligned} a_i + a_{i+n} &= b - 1 \\ a_1 \dots a_n + a_{n+1} \dots a_{2n} &= b^n - 1. \end{aligned}$$

这个定理还可再作推广（广义米迪定理）：若k是l的正因数，则 $a_1 a_2 \dots a_k + a_{k+1} a_{k+2} \dots a_{2k} + \dots + a_{l-k+1} a_{l-k+2} \dots a_l$ 是 $b^k - 1$ 的倍数。

例

$$\frac{1}{17} = 0.\overline{0588235294117647} \text{ and } 05882352 + 94117647 = 99999999.$$

$$\frac{1}{19} = 0.\overline{052631578947368421}$$

$$052631578 + 947368421 = 999999999$$

$$052631 + 578947 + 368421 = 999999$$

$$052 + 631 + 578 + 947 + 368 + 421 = 2997 = 3 \times 999.$$

$$\frac{1}{19} = 0.\overline{032745}_8$$

$$032_8 + 745_8 = 777_8$$

$$03_8 + 27_8 + 45_8 = 77_8.$$

【哥德巴赫猜想】

任一大于2的偶数都可写成两个素数之和；

任一大于7的奇数都可写成三个质数之和。

【沃尔斯滕霍尔姆定理】

在数论上，**Wolstenholme定理**说明，对于大于或等于5的质数，有

$$\binom{2p-1}{p-1} \equiv 1 \pmod{p^3},$$

$$\binom{ap}{bp} \equiv \binom{a}{b} \pmod{p^3},$$

$$(p-1)! \left(1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{p-1} \right) \equiv 0 \pmod{p^2},$$

$$(p-1)!^2 \left(1 + \frac{1}{2^2} + \frac{1}{3^2} + \dots + \frac{1}{(p-1)^2} \right) \equiv 0 \pmod{p}.$$

以上四个等式是等价的。

定理1 设 a, b 为整数, 且 $1 \leq a < b, (a, b) = 1$ 则 $\frac{a}{b}$ 可以化为有限小数的充要条件是: 分母 b 不含2和5之外的其他质因数, 当 $b = 2^\alpha \times 5^\beta$ 时, $\frac{a}{b}$ 是一个 s 位有限小数, 这里 $s = \max\{\alpha, \beta\}$.

定理2 设 $1 \leq a < b, (a, b) = 1$, 则 $\frac{a}{b}$ 可以化为纯循环

小数的充要条件是 $(b, 10) = 1$, 这时 $\frac{a}{b}$ 所化成的纯循环小数的循环节的长度 n 满足同余式 $10^n \equiv 1 \pmod{b}$ 的最小正整数.

定理3 设 $1 \leq a < b, (a, b) = 1$, 则 $\frac{a}{b}$ 可以化为混循环小数的充要条件是: 分母 b 含质因数2或5, 又含2和5之外其他质因数. 当 $b = 2^\alpha \times 5^\beta \times b_1$ (其中 $(b_1, 10) = 1$) 时, $\frac{a}{b}$ 所化成的混循环小数的不循环部分的长度为 $s = \max\{\alpha, \beta\}$ 循环节长度 n 是满足同余式 $10^n \equiv 1 \pmod{b_1}$ 的最小正整数。

勾股方程和佩尔方程

2018年10月25日 21:47

【勾股方程】

大家知道,能够使直角三角形三边的长为三个正整数的一组数,称为勾股数,用 (a, b, c) 表示,即 a 边为勾数, b 边为股数, c 边为弦数.在几何里,我们称 $a^2 + b^2 = c^2 (a < b)$ ①

为勾股定理,在代数里,我们称它为勾股方程.

两千多年前,古希腊人先后给出的几种勾股数计算公式都无法求到式①的每一个大于2的正整数 a 的一切正整数解.

2000年,笔者(张绍涛)又给出了新的 (a, b, c) 计算公式:

① a 为奇数的 (a, b, c) 计算公式是:

$$\begin{cases} a = 2n + 1, (n \text{ 为自然数}) \\ b = \frac{a^2 - j^2}{2j}, (j \text{ 是 } a^2 \text{ 中的因数}, 2j < a < b) \\ c = \frac{a^2 + j^2}{2j} \end{cases}$$

② a 为偶数的 (a, b, c) 计算公式是:

$$\begin{cases} a = 2n, (n \text{ 为正整数}) \\ b = \frac{a^2}{4j} - j, (j \text{ 是 } a^2 \text{ 中的因数}, 4j < a < b) \\ c = \frac{a^2}{4j} + j \end{cases}$$

许多数学工作者研究勾股定理的正整数解,他们推导出了求 $x^2 + y^2 = z^2$ 的正整数解(勾股数)的多种方法,这里我介绍一种求勾股数的一般通式.

预备定理1 自然数 x, y, z ,下面三种关系必定有一种且只有一种成立:(1) x, y, z 两两互质,(2) x, y, z 两两不互质,但 x, y, z 互质,(3) x, y, z 不互质.

预备定理2 若 x, y, z 两两互质, x 为正奇数, $x^2 = (z+y)(z-y)$,则 $(z+y)$ 与 $(z-y)$ 是互质的正奇数.

证明:假设 $(z+y)$ 与 $(z-y)$ 不是互质正奇数.

(1) 若 $(z+y)$ 与 $(z-y)$ 是互质的一奇一偶,则 $(z+y)(z-y)$ 是偶数,即 x 是偶数,这与假设 x 为正奇数矛盾.所以 $(z+y)$ 与 $(z-y)$ 不是互质的一奇一偶.

(2) 若 $(z+y)$ 与 $(z-y)$ 不互质,那么 $(z+y)$ 与 $(z-y)$ 有大于1的公因数 k ,可设 $z+y = ak, z-y = bk$.

$$\text{则 } x^2 = abk^2$$

所以 a, b, k 均为正奇数

所以 $a \pm b$ 是偶数

若 x 为正奇数,那么 $(z+y)$ 与 $(z-y)$ 必互质(预备定理2)

$$\text{设 } z+y = a^2, z-y = b^2,$$

$$\text{那么 } x^2 = (ab)^2,$$

$$\text{即 } x = ab \quad (a, b \text{ 为互质正奇数})$$

$$\text{解 } \begin{cases} z+y = a^2 \\ z-y = b^2 \end{cases} \text{ 得 } \begin{cases} y = \frac{1}{2}(a^2 - b^2) \\ z = \frac{1}{2}(a^2 + b^2) \end{cases}$$

即通解为:

$$\begin{cases} x = ab \\ y = \frac{1}{2}(a^2 - b^2) \\ z = \frac{1}{2}(a^2 + b^2) \end{cases}$$

(a, b 为互质正奇数 $a > b$) (I)

注:若设 y 为正奇数,通解中 x, y 互换即可.

(2) 若 x, y, z 两两不互质,但 x, y, z 互质.

设 x, y 有大于1的公约数,由于 $x^2 + y^2 = z^2$ 那么 z 也有这个约数.

$$\text{解} \begin{cases} z+y=ak \\ z-y=bk \end{cases} \text{得} \begin{cases} z=\frac{1}{2}(a+b)k \\ y=\frac{1}{2}(a-b)k \end{cases}$$

而 $\frac{1}{2}(a \pm b)$ 是整数.

所以 z 与 y 有大于 1 的公因数 k ,

这与已知 x, y, z 两两互质相矛盾,

所以 $(z+y)$ 与 $(z-y)$ 互质,

综上所述 $(z+y)$ 与 $(z-y)$ 是互质的正奇数.

定理: $x^2 + y^2 = z^2$ 的正整数解的通式为:

$$\begin{cases} x=kab \\ y=\frac{1}{2}k(a^2-b^2) \begin{cases} a, b \text{ 为互质正奇数} \\ a > b, k \in \mathbb{N} \end{cases} \\ z=\frac{1}{2}k(a^2+b^2) \end{cases}$$

推证: (1) 若 x, y, z 两两互质, 那么 x, y, z 中不存在两个偶数, x, y 中必有一个奇数.

由 $x^2 + y^2 = z^2$ 得 $x^2 = (z+y)(z-y)$

即求 $x^2 + y^2 = z^2$ 的正整数解(勾股数)的一般通式为:

$$\begin{cases} x=kab \\ y=\frac{1}{2}k(a^2-b^2) \begin{cases} a, b \text{ 为互质正奇数} \\ a > b, k \in \mathbb{N} \end{cases} \\ z=\frac{1}{2}k(a^2+b^2) \end{cases}$$

例 已知 $x=15$, 求适合方程 $x^2 + y^2 = z^2$ 的所有勾股数

解: 分解质因数得 $x=3 \times 5$, 按求勾股数的通式

则 x, y, z 就不互质了, 这与 x, y, z 互质相矛盾.

因此, 在 $x^2 + y^2 = z^2$ 中, x, y, z 两两不互质, 但 x, y, z 互质的情况不可能.

(3) 若 x, y, z 不互质, 则 x, y, z 有大于 1 的最大公约数 k ,

可设 $x=mk, y=nk, z=qk$ (m, n, q 互质)

i) 若 m, n, q 两两互质, 则可按情况(1)推得通解为:

$$\begin{cases} x=kab \\ y=\frac{1}{2}k(a^2-b^2) \begin{cases} a, b \text{ 为互质正奇数} \\ a > b, k > 1, k \in \mathbb{N} \end{cases} \\ z=\frac{1}{2}k(a^2+b^2) \end{cases} \quad (\text{II})$$

ii) 若 m, n, q 两两不互质, 但 m, n, q 互质, 则按情况(2)的分析在 $m^2 + n^2 = q^2$ 中, 此种情况不存在.

综上所述, $x^2 + y^2 = z^2$ 的正整数解的通式只有 (I)、(II) 两种情况, 而 (I)、(II) 两种通式可以统一成下列通式.

得:

$$k=1, a=5, b=3, x=15, y=8, z=17;$$

$$k=1, a=15, b=1, x=15, y=112, z=113;$$

$$k=3, a=5, b=1, x=15, y=36, z=39;$$

$$k=5, a=3, b=1, x=15, y=20, z=25.$$

共有四种勾股数:

$$\begin{cases} x=15 \\ y=8 \\ z=17 \end{cases} \begin{cases} x=15 \\ y=112 \\ z=113 \end{cases} \begin{cases} x=15 \\ y=36 \\ z=39 \end{cases} \begin{cases} x=15 \\ y=20 \\ z=25 \end{cases}$$

【佩尔方程】

定理 设 $D \in \mathbb{N}_+$, 且不是完全平方数.

则形如

$$x^2 - Dy^2 = 1 \quad (2)$$

的方程叫做佩尔方程.

如果 (x_1, y_1) 是使 x_1 最小的方程②的解(称为最小解),则每个解 (x_k, y_k) 都可以取幂得到

$$x_k + \sqrt{D}y_k = (x_1 + \sqrt{D}y_1)^k \quad (k \in \mathbf{N}_+). \quad (3)$$

如果我们求出佩尔方程的最小正整数解后,就可以根据递推式求出所有的解。

$$x_n = x_{n-1}x_1 + dy_{n-1}y_1$$

$$y_n = x_{n-1}y_1 + y_{n-1}x_1$$

则根据上式我们可以构造矩阵,然后就可以快速幂了。

$$\begin{bmatrix} x_k \\ y_k \end{bmatrix} = \begin{bmatrix} x_1 & dy_1 \\ y_1 & x_1 \end{bmatrix}^{k-1} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

这样就可以求出第k大的解。

HDU3292题就要用到上面的矩阵方法求第k大的解。

拓展一点:

(1) 如果第n个三角数t等于m的平方, 即 $\frac{n(n+1)}{2} = m^2$, 那么 $x=2n+1$, $y=m$, 就是丢番图方程 $x^2 - 8y^2 = 1$ 的解。

(2) 求丢番图方程 $x^2 - dy^2 = -1$ 的最小正整数解, 其中d为非完全平方数的正整数。

题目: <http://poj.org/problem?id=2427>

题意: 求方程 $x^2 - n \cdot y^2 = 1$ 的最小正整数解。本题要用到高精度, 所以用Java。

```
import java.math.BigInteger;
import java.util.Scanner;

public class Main
{
    public static void solve(int n)
```

```

    {
        BigInteger N, p1, p2, q1, q2, a0, a1, a2, g1, g2, h1,
h2, p, q;
        g1 = q2 = p1 = BigInteger.ZERO;
        h1 = q1 = p2 = BigInteger.ONE;
        a0 = a1 = BigInteger.valueOf((int)Math.sqrt(1.0*n));
        BigInteger ans=a0.multiply(a0);
        if(ans.equals(BigInteger.valueOf(n)))
        {
            System.out.println("No solution!");
            return;
        }
        N = BigInteger.valueOf(n);
        while (true)
        {
            g2 = a1.multiply(h1).subtract(g1);
            h2 = N.subtract(g2.pow(2)).divide(h1);
            a2 = g2.add(a0).divide(h2);
            p = a1.multiply(p2).add(p1);
            q = a1.multiply(q2).add(q1);
            if
(p.pow(2).subtract(N.multiply(q.pow(2))).compareTo(BigInteger.ONE
E) == 0) break;
            g1 = g2;h1 = h2;a1 = a2;
            p1 = p2;p2 = p;
            q1 = q2;q2 = q;
        }
        System.out.println(p+" "+q);
    }

    public static void main(String[] args)
    {
        Scanner cin = new Scanner(System.in);
        while(cin.hasNextInt())
        {
            solve(cin.nextInt());
        }
    }
}

```

题意：给出一个数N，求1到N的范围内，找到一个最大的n，满足 $6x^2 = (n+1)(2n+1)$ ，N最大达到 10^{18}

分析：我们把上式写成 $(4n+3)^2 - 48x^2 = 1$ ，然后就是解Pell方程即可。

基本组合定理

2018年10月25日 11:21

【鸽巢原理】

第一抽屉原理

原理1

把多余 $n+1$ 个物体放到 n 个抽屉里，则至少有一个抽屉里的东西不少于两件。

原理2

把多余 $mn+1$ (n 不为0)个物体放到 n 个抽屉里面，则至少有一个抽屉里面不少于 $(m+1)$ 的物体。

第二抽屉原理

把 $(mn - 1)$ 个物体放入 n 个抽屉中，其中必须有一个抽屉不多余 $(m-1)$ 个物体。

【ramsey定理】

要找这样一个最小的数 n ，使得 n 个人中必定有 k 个人相识或1个人互不相识。

Ramsey定理：对于一个给定的两个整数 $m, n \geq 2$ ，则一定存在一个最小整数 r ，使得用两种颜色（例如红蓝）无论给 K_r 的每条边如何染色，总能找到一个红色的 K_m 或者蓝色的 K_n

Example: $R(3, 3) = 6$

【cayley定理】

n 个有标号点完全图的生成树个数有 n^{n-2} 个

【矩阵树定理】

矩阵树定理 Kirchhoff Matrix-Tree

1. G 的度数矩阵 $D[G]$ 是一个 $n * n$ 的矩阵当 $i \neq j$ 时, $d_{ij} = 0$; 当 $i = j$ 时, d_{ij} 等于 i 的度数;

2. G 的邻接矩阵 $A[G]$;

我们定义 G 的Kirchhoff矩阵(也称为拉普拉斯算子) $C[G] = D[G] - A[G]$, 则Matrix-Tree定理可以描述为:

G 的所有不同的生成树的个数等于其Kirchhoff矩阵 $C[G]$ 任何一个 $n - 1$ 阶主子式的行列式的绝对值。

【容斥原理】

$$|A_1 \cup A_2 \cup \dots \cup A_m| =$$

$$\sum_{1 \leq i \leq m} |A_i| - \sum_{1 \leq i < j \leq m} |A_i \cap A_j| + \sum_{1 \leq i < j < k \leq m} |A_i \cap A_j \cap A_k| - \dots + (-1)^{m-1} |A_1 \cap A_2 \cap \dots \cap A_m|$$

【四色定理】

任何一张地图只用四种颜色就能使具有共同边界的国家着上不同的颜色。

【五色定理】

五色定理是图论中的一个结论：将一个平面分成若干区域，给这些区域染色，且保证任意相邻区域没有相同颜色，那么所需颜色不超过五种。

【图兰定理】

Turán's Theorem. Let G be any graph with n vertices, such that G is K_{r+1} -free. Then the number of edges in G is at most

$$\frac{r-1}{r} \cdot \frac{n^2}{2} = \left(1 - \frac{1}{r}\right) \cdot \frac{n^2}{2}.$$

An equivalent formulation is the following:

Turán's Theorem (Second Formulation). Among the n -vertex simple graphs with no $(r+1)$ -cliques, $T(n, r)$ has the maximum number of edges.

Mantel's Theorem. The maximum number of edges in an n -vertex triangle-free graph is $\lfloor n^2/4 \rfloor$.

各种反演公式

2018年10月25日 13:00

【莫比乌斯函数反演】

$$f(x) = \sum_{d|x} g(d) \Rightarrow g(x) = \sum_{d|x} \mu(d) * f\left(\left\lfloor \frac{n}{d} \right\rfloor\right)$$

$$f(x) = \sum_{x|d} g(d) \Rightarrow g(x) = \sum_{x|d} \mu\left(\left\lfloor \frac{d}{n} \right\rfloor\right) * f(d)$$

【二项式反演】

$$f_n = \sum_{i=0}^n (-1)^i \binom{n}{i} g_i \Leftrightarrow g_n = \sum_{i=0}^n (-1)^i \binom{n}{i} f_i$$

$$f_n = \sum_{i=0}^n \binom{n}{i} g_i \Leftrightarrow g_n = \sum_{i=0}^n (-1)^{n-i} \binom{n}{i} f_i$$

这是关于下指标反演，同样也可以关于上指标反演

【Stirling反演】

$$f(n) = \sum_{i=1}^n \left\{ \begin{matrix} n \\ i \end{matrix} \right\} g(i)$$

$$g(n) = \sum_{i=1}^n (-1)^{n-i} \left[\begin{matrix} n \\ i \end{matrix} \right] f(i)$$

【min/max反演】

$$\max(S) = \sum_{T \subseteq S} (-1)^{|T|-1} * \min(T)$$

$$\min(S) = \sum_{T \subseteq S} (-1)^{|T|-1} * \max(T)$$

【拉格朗日反演】

$$[z^n]g(z) = \frac{1}{n}[w^{n-1}]\phi(w)^n,$$

$$[z^n]H(g(z)) = \frac{1}{n}[w^{n-1}](H'(w)\phi(w)^n)$$

$$[z^n]H(g(z)) = [w^n]H(w)\phi(w)^{n-1}(\phi(w) - w\phi'(w)),$$

组合数的简单推广

2018年10月25日 13:21

$$\binom{n}{k_1, k_2, \dots, k_r} = \frac{n!}{k_1! k_2! \cdots k_r!}$$

$$\binom{1/2}{k} = \binom{2k}{k} \frac{(-1)^{k+1}}{2^{2k} (2k-1)}.$$

$$\binom{n}{k} = (-1)^k \binom{k-n-1}{k}$$

$$\begin{aligned} \binom{-n}{k} &= \frac{-n \cdot -(n+1) \cdots -(n+k-2) \cdot -(n+k-1)}{k!} \\ &= (-1)^k \frac{n \cdot (n+1) \cdot (n+2) \cdots (n+k-1)}{k!} \\ &= (-1)^k \binom{n+k-1}{k} \\ &= (-1)^k \left(\binom{n}{k} \right). \end{aligned}$$

五边形数定理和划分数

2018年10月25日 19:52

$$\prod_{n=1}^{\infty} (1 - x^n) = \sum_{k=-\infty}^{\infty} (-1)^k x^{k(3k-1)/2} = 1 + \sum_{k=1}^{\infty} (-1)^k \left(x^{k(3k+1)/2} + x^{k(3k-1)/2} \right).$$

$$p(n) = \sum_k (-1)^{k-1} p(n - g_k)$$

where the summation is over all nonzero integers k (positive and negative) and g_k is the k^{th} generalized pentagonal number. Since $p(n)$ is zero for all $n < 0$, the series will eventually become zeroes, enabling discrete calculation.

$$\sum_{n=0}^{\infty} p(n) x^n = \prod_{k=1}^{\infty} (1 - x^k)^{-1}$$

Note that is the reciprocal of the product on the left hand side of our identity:

$$\left(\sum_{n=0}^{\infty} p(n) x^n \right) \cdot \left(\prod_{n=1}^{\infty} (1 - x^n) \right) = 1$$

Let us denote the expansion of our product by $\prod_{n=1}^{\infty} (1 - x^n) = \sum_{n=0}^{\infty} a_n x^n$, so that

$$\left(\sum_{n=0}^{\infty} p(n) x^n \right) \cdot \left(\sum_{n=0}^{\infty} a_n x^n \right) = 1.$$

Multiplying out the left hand side and equating coefficients on the two sides, we obtain $a_0 p(0) = 1$ and

$\sum_{i=0}^n p(n-i) a_i = 0$ for all $n \geq 1$. This gives a recurrence relation defining $p(n)$ in terms of a_n and vice versa a

recurrence for a_n in terms of $p(n)$. Thus, our desired result:

$$a_i := \begin{cases} 1 & \text{if } i = \frac{1}{2}(3k^2 \pm k) \text{ and } k \text{ is even} \\ -1 & \text{if } i = \frac{1}{2}(3k^2 \pm k) \text{ and } k \text{ is odd} \\ 0 & \text{otherwise} \end{cases}$$

for $i \geq 1$ is equivalent to the identity $\sum_i (-1)^i p(n - g_i) = 0$, where $g_i := \frac{1}{2}(3i^2 - i)$ and i ranges over all

integers such that $g_i \leq n$ (this range includes both positive and negative i , so as to use both kinds of generalized pentagonal numbers). This in turn means:

$$\sum_{i \text{ even}} p(n - g_i) = \sum_{i \text{ odd}} p(n - g_i),$$

In terms of sets of partitions, this is equivalent to saying that the following sets are of equal cardinality:

$$\mathcal{X} := \bigcup_{i \text{ even}} \mathcal{P}(n - g_i) \quad \text{and} \quad \mathcal{Y} := \bigcup_{i \text{ odd}} \mathcal{P}(n - g_i),$$

Other recurrence relations [edit]

A recurrence relation for $p(n)$ can be given in terms of the [sum of divisors function](#) σ :^[*citation needed*]

$$p(n) = \frac{1}{n} \sum_{k=0}^{n-1} \sigma(n-k)p(k).$$

If $q(n)$ denotes the number of partitions of n with no repeated parts then also^[*citation needed*]

$$p(n) = \sum_{k=0}^{\lfloor n/2 \rfloor} q(n-2k)p(k).$$

五边形数测试：

利用以下的公式可以测试一个正整数x是否是五边形数（此处不考虑广义五边形数）

$$n = \frac{\sqrt{24x+1}+1}{6}.$$

若n是自然数，则x是五边形数，而且恰为第n个五边形数。

若n不是自然数，则x不是五边形数。

【总结】

五边形数：0，1，2，5，7，12，15，22，26，35....

对应下标：0，1，-1，2，-2，3，-3，4，-4，5.....

$$p(k) = p(k-1) + p(k-2) - p(k-5).....$$

Hdu 4658 要求拆分的数中每个数出现的次数不能大于等于k次，则：

$$P_k(x) = (1+x+x^2+...+x^{k-1})(1+x^2+x^4+...+x^{2(k-1)})....$$

$$= \prod_{i=1}^{+\infty} \sum_{n=0}^{k-1} x^{ni} = \prod_{i=1}^{+\infty} \frac{1-x^{ik}}{1-x^i} = \frac{Q(x^k)}{Q(x)} = Q(x^k)P(x)$$

$P(x)$ 已经求得，现在看 $Q(x^k)$ 会怎么样

$$\begin{aligned} Q(x^k) &= \prod_{n=1}^{+\infty} (1-x^n) = \sum_{z=-\infty}^{+\infty} (-1)^z (x^k)^{z(3z-1)/2} \\ &= 1 - x^k - x^{2k} + x^{5k} + x^{7k} - x^{22k} \dots \end{aligned}$$

例如，当 $n=8$ ， $k=4$ 时

$$Q(x^4)P(x) = (1-x^4-x^8+x^{12}....)(1+x^1+2x^2+3x^3+5x^4+7x^5+11x^6+15x^7+22x^8+30x^9+...)$$

满足指数为 8 的乘积之和为： $1*22x^8 - x^4*5x^4 - x^8*1 = 16x^8$

简要模板:

```
LL p[MAXN];
void init() {
    p[0] = p[1] = 1;
    p[2] = 2;
    p[3] = 3;
    for(int i = 4; i <= 100000; i++)
    {
        p[i] = 0;
        int flag = 1;
        for(int j = 1; ; j++)
        {
            int p1 = j * (3 * j - 1) / 2;
            int p2 = j * (3 * j + 1) / 2;
            if(p1 > i && p2 > i) break;
            if(p1 <= i) p[i] = (p[i] + flag * p[i - p1] + MOD) %
MOD;
            if(p2 <= i) p[i] = (p[i] + flag * p[i - p2] + MOD) %
MOD;
            flag *= (-1);
        }
    }
}

LL solve(int n, int k)
{
    LL ret = p[n];
    int flag = -1;
    for(int j = 1; ; j++)
    {
        int p1 = k * j * (3 * j - 1) / 2;
        int p2 = k * j * (3 * j + 1) / 2;
        if(p1 > n && p2 > n) break;
        if(p1 <= n) ret = (ret + flag * p[n - p1] + MOD) % MOD;
        if(p2 <= n) ret = (ret + flag * p[n - p2] + MOD) % MOD;
        flag *= (-1);
    }
    return ret;
}

int main()
{
    init();
    int t, n, k;
```



```
scanf("%d", &t);  
while(t--)  
{  
    scanf("%d%d", &n, &k);  
    printf("%I64d\n", solve(n, k));  
}  
return 0;  
}
```

prufer编码

2018年10月25日 21:12

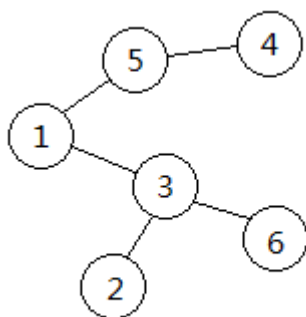
Prufer数列是无根树的一种数列。在组合数学中，Prufer数列由有一个对于顶点标过号的树转化来的数列，点数为 n 的树转化来的Prufer数列长度为 $n-2$ 。它可以通过简单的迭代方法计算出来。

办法

将树转化成Prufer数列的方法

一种生成Prufer序列的方法是迭代删点，直到原图仅剩两个点。对于一棵顶点已经经过编号的树 T ，顶点的编号为 $\{1, 2, \dots, n\}$ ，在第 i 步时，移去所有叶子节点（度为1的顶点）中标号最小的顶点和相连的边，并把与它相邻的点的编号加入Prufer序列中，重复以上步骤直到原图仅剩2个顶点。

例子



Prufer数列

以右边的树为例子，首先在所有叶子节点中编号最小的点是2，和它相邻的点的编号是3，将3加入序列并删除编号为2的点。接下来删除的点是4，5被加入序列，然后删除5，1被加入序列，1被删除，3被加入序列，此时原图仅剩两个点（即3和6），Prufer序列构建完成，为 $\{3, 5, 1, 3\}$

将Prufer数列转化成树的方法

设 $\{a_1, a_2, \dots, a_{n-2}\}$ 为一棵有 n 个节点的树的Prufer序列，另建一个集合 G 含有元素 $\{1..n\}$ ，找出集合中最小的未在Prufer序列中出现过的数，将该点与Prufer序列中首项连一条边，并将该点和Prufer序列首项删除，重复操作 $n-2$ 次，将集合中剩余的两个点之间连边即可。

例子

仍为上面的树，Prufer序列为 $\{3, 5, 1, 3\}$ ，开始时 $G = \{1, 2, 3, 4, 5, 6\}$ ，未出现的编号最小的点是2，将2和3连边，并删去Prufer序列首项和 G 中的2。接下来连的边为 $\{4, 5\}, \{1, 5\}, \{1, 3\}$ ，此时集合 G 中仅剩3和6，在3和6之间连边，原树恢复。

库默尔定理

2018年10月25日 21:27

设 m, n 为正整数, p 为素数, 则 $C(m+n, m)$ 含 p 的幂次等于 $m+n$ 在 p 进制下的进位次数。

组合数 $\binom{m+n}{m}$ 所含 p 的幂次数为

$$\sum_{i=1}^{\infty} \left[\frac{m+n}{p^i} \right] - \sum_{i=1}^{\infty} \left[\frac{n}{p^i} \right] - \sum_{i=1}^{\infty} \left[\frac{m}{p^i} \right]$$
$$= \sum_{i=1}^{\infty} \left(\left[\frac{m+n}{p^i} \right] - \left[\frac{n}{p^i} \right] - \left[\frac{m}{p^i} \right] \right)$$

这是因为组合数公式 $\binom{m+n}{m} = \frac{(m+n)!}{m!n!}$ 以及 $n!$ 含有素数 p 的幂次公式 $v_p(n!) = \sum_{i=1}^{\infty} \left[\frac{n}{p^i} \right]$ 。

对于某个 p^i , $\left[\frac{m}{p^i} \right]$ 等于 m 在 p 进制表示下去掉后 i 位, 在第 $i+1$ 位上, $m+n$ 在这一位上进位的充要条件是 $\left[\frac{m+n}{p^i} \right] - \left[\frac{n}{p^i} \right] - \left[\frac{m}{p^i} \right] = 1$, 不进位则 $\left[\frac{m+n}{p^i} \right] - \left[\frac{n}{p^i} \right] - \left[\frac{m}{p^i} \right] = 0$. 因此 $\sum_{i=1}^{\infty} \left(\left[\frac{m+n}{p^i} \right] - \left[\frac{n}{p^i} \right] - \left[\frac{m}{p^i} \right] \right)$ 就是 $m+n$ 在 p 进制下的进位次数。

例 (2014 CMO [1] 30, 4, 21分) 求具有下述性质的所有整数 k : 存在无穷多个正整数 n , 使得 $n+k$ 不整除 $\binom{2n}{n}$ 。

解 $\because \binom{2n}{n-1} = \frac{2n(2n-1)\cdots(n+2)}{(n-1)!} = \frac{n}{n+1} \binom{2n}{n},$

$\therefore \frac{1}{n+1} \binom{2n}{n} = \binom{2n}{n} - \binom{2n}{n-1}$ 是整数,

$\therefore n+1 \mid \binom{2n}{n}$ 对任意正整数 n 成立, 从而1不满足要求。

当 $k \leq 0$ 时, 取 $n=p-k$ (p 为奇素数, $p > -2k$), 满足要求。

当 $k \geq 2$ 时, 取 k 的一个素因子 p , 选取正整数 m 使得 $p^m > k$, 令 $n=p^m-k$, 我们证明: $n+k$ 不整除 $\binom{2n}{n}$ 。

显然有 $n > 0$, 由 $n < p^m$ 知 n 在 p 进制下最多 n 位, $\therefore p \nmid k, p \mid p^m, \therefore p \nmid n$ 。 \therefore 在 $p+1$ 进制下 n 个位为0。

$\therefore 2n=n+n$ 最多进位 $m-1$ 次。由库默尔定理, $\binom{2n}{n}$ 最多有 $m-1$ 个 p , $\therefore n+k=p^m, \therefore$

康托展开与其逆

2018年10月25日 21:30

康托展开运算

$$X = a_n(n-1)! + a_{n-1}(n-2)! + \dots + a_1 \cdot 0!$$

其中, a_i 为整数, 并且 $0 \leq a_i < i, 1 \leq i \leq n$ 。

康托展开举例

再举个例子说明。

在 (1, 2, 3, 4, 5) 5 个数的排列组合中, 计算 34152 的康托展开值。

首位是 3, 则小于 3 的数有两个, 为 1 和 2, $a[5] = 2$, 则首位小于 3 的所有排列组合为 $a[5] \times (5-1)!$

第二位是 4, 则小于 4 的数有两个, 为 1 和 2, 注意这里 3 并不能算, 因为 3 已经在第一位, 所以其实计算的是在第二位之后小于 4 的个数。因此 $a[4] = 2$ 。

第三位是 1, 则在其之后小于 1 的数有 0 个, 所以 $a[3] = 0$ 。

第四位是 5, 则在其之后小于 5 的数有 1 个, 为 2, 所以 $a[2] = 1$ 。

最后一位就不用计算啦, 因为在它之后已经没有数了, 所以 $a[1]$ 固定为 0

根据公式:

$$X = 2 \times 4! + 2 \times 3! + 0 \times 2! + 1 \times 1! + 0 \times 0! = 61$$

所以比 34152 小的组合有 61 个, 即 34152 是排第 62。

代码实现

```
1 static const int FAC[] = {1, 1, 2, 6, 24, 120, 720, 5040, 40320, 362880}; // 阶乘
2 int cantor(int *a, int n)
3 {
4     int x = 0;
5     for (int i = 0; i < n; ++i) {
6         int smaller = 0; // 在当前位之后小于其的个数
7         for (int j = i + 1; j < n; ++j) {
8             if (a[j] < a[i])
9                 smaller++;
10        }
11        x += FAC[n - i - 1] * smaller; // 康托展开累加
12    }
13    return x; // 康托展开值
14 }
```

逆康托展开举例

一开始已经提过了，康托展开是一个全排列到一个自然数的双射，因此是可逆的。即对于上述例子，在 $(1, 2, 3, 4, 5)$ 给出61可以算出起排列组合为34152。由上述的计算过程可以容易的逆推回来，具体过程如下：

用 $61 / 4! = 2$ 余 13，说明 $a[5] = 2$ ，说明比首位小的数有2个，所以首位为3。

用 $13 / 3! = 2$ 余 1，说明 $a[4] = 2$ ，说明在第二位之后小于第二位的数有2个，所以第二位为4。

用 $1 / 2! = 0$ 余 1，说明 $a[3] = 0$ ，说明在第三位之后没有小于第三位的数，所以第三位为1。

用 $1 / 1! = 1$ 余 0，说明 $a[2] = 1$ ，说明在第二位之后小于第四位的数有1个，所以第四位为5。

最后一位自然就是剩下的数2。

通过以上分析，所求排列组合为 34152。

代码实现

```
1 static const int FAC[] = {1, 1, 2, 6, 24, 120, 720, 5040, 40320, 362880}; // 阶乘
2
3 //康托展开逆运算
4 void decantor(int x, int n)
5 {
6     vector<int> v; // 存放当前可选数
7     vector<int> a; // 所求排列组合
8     for(int i=1;i<=n;i++)
9         v.push_back(i);
10    for(int i=n;i>=1;i--)
11    {
12        int r = x % FAC[i-1];
13        int t = x / FAC[i-1];
14        x = r;
15        sort(v.begin(),v.end()); // 从小到大排序
16        a.push_back(v[t]); // 剩余数里第t+1个数当前位
17        v.erase(v.begin()+t); // 移除选做当前位的数
18    }
19 }
```

约瑟夫环

2018年10月25日 22:11

【k=2特例】

把n写成二进制，循环左移一位，就是答案

【一般情况】

编号1-n:

$$f(n, k) = ((f(n-1, k) + k - 1) \bmod n) + 1, \text{ with } f(1, k) = 1,$$

编号0-n-1:

$$g(n, k) = (g(n-1, k) + k) \bmod n, \text{ with } g(1, k) = 0$$

This approach has [running time](#) $O(n)$, but for small k and large n there is another approach. The second approach also uses dynamic programming but has running time $O(k \log n)$. It is based on considering killing k -th, $2k$ -th, ..., $(\lfloor n/k \rfloor k)$ -th people as one step, then changing the numbering.^{[\[citation needed\]](#)}

This improved approach takes the form

$$g(n, k) = \begin{cases} 0 & \text{if } n = 1 \\ (g(n-1, k) + k) \bmod n & \text{if } 1 < n < k \\ \left\lfloor \frac{k(g(n', k) - n \bmod k) \bmod n'}{k-1} \right\rfloor \text{ where } n' = n - \left\lfloor \frac{n}{k} \right\rfloor & \text{if } k \leq n \end{cases}$$

$[n]$ 中 k 元子集元素乘积之和

2018年10月25日 22:22

对于前 n 个正整数构成的集合，求所有 k 元子集元素乘积之和（ $1 \leq k \leq n$ 中 k 元子集元素乘积之和）

答案： $ans = S[n+1][n-k+1]$, S 表示第一类斯特林数

有向欧拉回路计数：BEST定理

2018年10月25日 22:34

In graph theory, a part of discrete mathematics, the BEST theorem gives a product formula for the number of Eulerian circuits in directed (oriented) graphs.

Precise statement [\[edit\]](#)

Let $G = (V, E)$ be a directed graph. An Eulerian circuit is a directed closed path which visits each edge exactly once. In 1736, [Euler](#) showed that G has an Eulerian circuit if and only if G is [connected](#) and the [indegree](#) is equal to [outdegree](#) at every vertex. In this case G is called Eulerian. We denote the in-degree of a vertex v by $\deg(v)$.

The BEST theorem states that the number $\text{ec}(G)$ of Eulerian circuits in a connected Eulerian graph G is given by the formula

$$\text{ec}(G) = t_w(G) \prod_{v \in V} (\deg(v) - 1)!$$

Here $t_w(G)$ is the number of [arborescences](#), which are [trees](#) directed towards the root at a fixed vertex w in G . The number $t_w(G)$ can be computed as a [determinant](#), by the version of the [matrix tree theorem](#) for directed graphs. It is a property of Eulerian graphs that $t_v(G) = t_w(G)$ for every two vertices v and w in a connected Eulerian graph G .

Applications [\[edit\]](#)

The BEST theorem shows that the number of Eulerian circuits in directed graphs can be computed in [polynomial time](#), a problem which is [#P-complete](#) for undirected graphs.^[1] It is also used in the asymptotic enumeration of Eulerian circuits of [complete](#) and [complete bipartite graphs](#).^{[2][3]}

polya计数法

2018年10月25日 22:46

$$|Y^X/G| = \frac{1}{|G|} \sum_{g \in G} m^{c(g)}$$

where $m = |Y|$ is the number of colors and $c(g)$ is the number of [cycles](#) of the group element g when considered as a permutation of X .

Full, weighted version [\[edit \]](#)

In the more general and more important version of the theorem, the colors are also weighted in one or more ways, and there could be an infinite number of colors provided that the set of colors has a [generating function](#) with finite coefficients. In the univariate case, suppose that

$$f(t) = f_0 + f_1 t + f_2 t^2 + \dots$$

is the generating function of the set of colors, so that there are f_w colors of weight w for each integer $w \geq 0$. In the multivariate case, the weight of each color is a vector of integers and there is a generating function $f(t_1, t_2, \dots)$ that tabulates the number of colors with each given vector of weights.

The enumeration theorem employs another multivariate generating function called the [cycle index](#):

$$Z_G(t_1, t_2, \dots, t_n) = \frac{1}{|G|} \sum_{g \in G} t_1^{j_1(g)} t_2^{j_2(g)} \dots t_n^{j_n(g)}$$

where n is the number of elements of X and $j_k(g)$ is the number of k -cycles of the group element g as a permutation of X .

A colored arrangement is an orbit of the action of G on the set Y^X (where Y is the set of colors and Y^X denotes the set of all functions $\varphi: X \rightarrow Y$). The *weight* of such an arrangement is defined as the sum of the weights of $\varphi(x)$ over all $x \in X$. The theorem states that the generating function F of the number of colored arrangements by weight is given by:

$$F(t) = Z_G(f(t), f(t^2), f(t^3), \dots, f(t^n))$$

or in the multivariate case:

$$F(t_1, t_2, \dots) = Z_G(f(t_1, t_2, \dots), f(t_1^2, t_2^2, \dots), f(t_1^3, t_2^3, \dots), \dots, f(t_1^n, t_2^n, \dots)).$$

To reduce to the simplified version given earlier, if there are m colors and all have weight 0, then $f(t) = m$ and

$$|Y^X/G| = F(0) = Z_G(m, m, \dots, m) = \frac{1}{|G|} \sum_{g \in G} m^{c(g)}.$$

In the celebrated application of counting trees (see below) and acyclic molecules, an arrangement of "colored beads" is actually an arrangement of arrangements, such as branches of a rooted tree. Thus the generating function f for the colors is derived from the generating function F for arrangements, and the Pólya enumeration theorem becomes a recursive formula.

In the celebrated application of counting trees (see below) and acyclic molecules, an arrangement of "colored beads" is actually an arrangement of arrangements, such as branches of a rooted tree. Thus the generating function f for the colors is derived from the generating function F for arrangements, and the Pólya enumeration theorem becomes a recursive formula.

Colored cubes [\[edit \]](#)

How many ways are there to color the sides of a three-dimensional cube with m colors, up to rotation of the cube? The rotation group C of the cube acts on the six sides of the cube, which are equivalent to beads. Its cycle index is

$$Z_C(t_1, t_2, t_3, t_4) = \frac{1}{24} (t_1^6 + 6t_1^2t_4 + 3t_1^2t_2^2 + 8t_3^2 + 6t_2^3)$$

which is obtained by analyzing the action of each of the 24 elements of C on the 6 sides of the cube, see [here](#) for the details.

We take all colors to have weight 0 and find that there are

$$F(0) = Z_C(m, m, m, m) = \frac{1}{24} (m^6 + 3m^4 + 12m^3 + 8m^2)$$

different colorings.

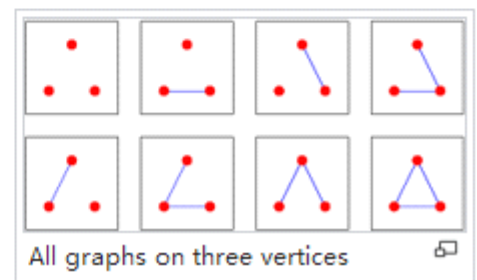
Graphs on three and four vertices [\[edit \]](#)

A graph on m vertices can be interpreted as an arrangement of colored beads. The set X of "beads" is the set of $\binom{m}{2}$ possible edges, while the set of colors $Y = \{\text{black, white}\}$

corresponds to edges that are present (black) or absent (white). The Pólya enumeration theorem can be used to calculate the number of graphs [up to isomorphism](#) with a fixed number of vertices, or the generating function of these graphs according to the number of edges they have. For the latter purpose, we can say that a black or present edge has weight 1, while an absent or white edge has weight 0. Thus $f(t) = 1 + t$ is the generating function for the set of colors. The relevant symmetry group is $G = S_m$, the [symmetric group](#) on m letters. This group acts on the set X of possible edges: a permutation φ turns the edge $\{a, b\}$ into the edge $\{\varphi(a), \varphi(b)\}$. With these definitions, an isomorphism class of graphs with m vertices is the same as an orbit of the action of G on the set Y^X of colored arrangements; the number of edges of the graph equals the weight of the arrangement.

The eight graphs on three vertices (before identifying isomorphic graphs) are shown at the right. There are four isomorphism classes of graphs, also shown at the right.

The cycle index of the group S_3 acting on the set of three edges is



$$Z_G(t_1, t_2, t_3) = \frac{1}{6} (t_1^3 + 3t_1t_2 + 2t_3)$$

(obtained by inspecting the cycle structure of the action of the group elements; see [here](#)). Thus, according to the enumeration theorem, the generating function of graphs on 3 vertices up to isomorphism is

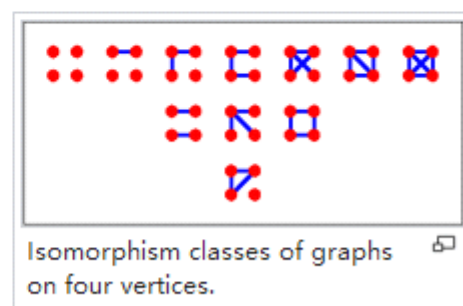
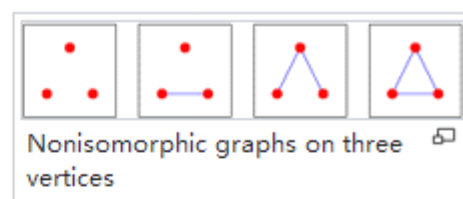
$$F(t) = Z_G(t+1, t^2+1, t^3+1) = \frac{1}{6} ((t+1)^3 + 3(t+1)(t^2+1) + 2(t^3+1))$$

which simplifies to

$$F(t) = t^3 + t^2 + t + 1.$$

Thus there is one graph each with 0 to 3 edges.

The cycle index of the group S_4 acting on the set of 6 edges is



$$Z_G(t_1, t_2, t_3, t_4) = \frac{1}{24} (t_1^6 + 9t_1^2t_2^2 + 8t_3^2 + 6t_2t_4)$$

$$F(t) = Z_G(t+1, t^2+1, t^3+1, t^4+1) =$$

$$\frac{(t+1)^6 + 9(t+1)^2(t^2+1)^2 + 8(t^3+1)^2 + 6(t^2+1)(t^4+1)}{24}$$

which simplifies to

$$F(t) = t^6 + t^5 + 2t^4 + 3t^3 + 2t^2 + t + 1.$$

These graphs are shown at the right.

Rooted ternary trees [\[edit \]](#)

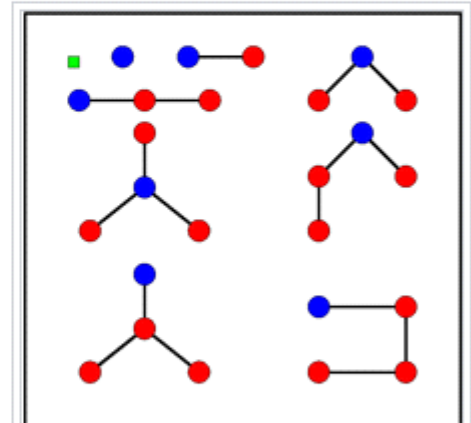
The set T_3 of rooted ternary trees consists of rooted trees where every node (or non-leaf vertex) has exactly three children (leaves or subtrees). Small ternary trees are shown at right. Note that rooted ternary trees with n nodes are equivalent to rooted trees with n vertices of degree at most 3 (by ignoring the leaves). In general, two rooted trees are isomorphic when one can be obtained from the other by permuting the children of its nodes. In other words, the group that acts on the children of a node is the symmetric group S_3 . We define the weight of such a ternary tree to be the number of nodes (or non-leaf vertices).

One can view a rooted, ternary tree as a recursive object which is either a leaf or a node with three children which are themselves rooted ternary trees. These children are equivalent to beads; the cycle index of the symmetric group S_3 that acts on them is

$$Z_{S_3}(t_1, t_2, t_3) = \frac{t_1^3 + 3t_1t_2 + 2t_3}{6}.$$

The Pólya enumeration theorem translates the recursive structure of rooted ternary trees into a functional equation for the generating function $F(t)$ of rooted ternary trees by number of nodes. This is achieved by "coloring" the three children with rooted ternary trees, weighted by node number, so that the color generating function is given by

$$f(t) = F(t)$$



Rooted ternary trees on 0, 1, 2, 3 and 4 nodes (=non-leaf vertices). The root is shown in blue, the leaves are not shown. Every node has as many leaves as to make the number of its children equal to 3.

which by the enumeration theorem gives

$$\frac{F(t)^3 + 3F(t)F(t^2) + 2F(t^3)}{6}$$

as the generating function for rooted ternary trees, weighted by one less than the node number (since the sum of the children weights does not take the root into account), so that

$$F(t) = 1 + t \frac{F(t)^3 + 3F(t)F(t^2) + 2F(t^3)}{6}.$$

This is equivalent to the following recurrence formula for the number t_n of rooted ternary trees with n nodes:

$$t_0 = 1$$

and

$$t_{n+1} = \frac{1}{6} \left(\sum_{a+b+c=n} t_a t_b t_c + 3 \sum_{a+2b=n} t_a t_b + 2 \sum_{3a=n} t_a \right)$$

where a , b and c are nonnegative integers.

The first few values of t_n are

1, 1, 1, 2, 4, 8, 17, 39, 89, 211, 507, 1238, 3057, 7639, 19241 (sequence [A000598](#) in the [OEIS](#))

abel求和变换

2018年10月25日 11:21

【abel求和变换公式】

$$\begin{aligned}\sum_{k=0}^n f_k g_k &= f_0 \sum_{k=0}^n g_k + \sum_{j=0}^{n-1} (f_{j+1} - f_j) \sum_{k=j+1}^n g_k \\ &= f_n \sum_{k=0}^n g_k - \sum_{j=0}^{n-1} (f_{j+1} - f_j) \sum_{k=0}^j g_k,\end{aligned}$$

$$S_N = \sum_{n=0}^N a_n b_n$$

$$S_N = a_N B_N - \sum_{n=0}^{N-1} B_n (a_{n+1} - a_n).$$

拉格朗日插值公式

2018年10月25日 16:52

对某个 k 次多项式 $f^k(n)$ 函数，已知有给定的 $k+1$ 个点：

$$(x_0, y_0), \dots, (x_k, y_k)$$

假设任意两个不同的 x_j 都互不相同，那么应用拉格朗日插值公式所得到的拉格朗日插值多项式为：

$$L(x) = \sum_{j=0}^k y_j l_j(x)$$

其中每个 $l_j(x)$ 为拉格朗日基本多项式（或称插值基函数），其表达式为：

$$l_j(x) := \prod_{i=0, i \neq j}^k \frac{x-x_i}{x_j-x_i} = \frac{(x-x_0)}{(x_j-x_0)} \cdots \frac{(x-x_{j-1})}{(x_j-x_{j-1})} \frac{(x-x_{j+1})}{(x_j-x_{j+1})} \cdots \frac{(x-x_k)}{(x_j-x_k)}.$$

拉格朗日基本多项式 $l_j(x)$ 的特点是在 x_j 上取值为1，在其它的点 $x_i, i \neq j$ 上取值为0。

这样对于每个 $L(x)$ ，它必定过所有 $(x_0, y_0), \dots, (x_k, y_k)$

在实际的OI竞赛中，选手常常要自己带值插值，所以我们一般选择 $(0, f^k(0)), \dots, (k, f^k(k))$ ，带入 $L(x)$ 有 $f^k(n) = L(n) = \sum_{j=0}^{k+1} \frac{f^k(j)n(n-1)\dots(n-j+1)(n-j-1)\dots(n-k-1)(-1)^{k+1-j}}{j!(k+1-j)!}$

于是我们可以预处理阶乘等来使得时间复杂度变成 $O(k)$ 的

缺点就是对模数有要求。

```
1 long long Lagrange(long long *a, int n, long long pos) {
2     if(pos <= n) return a[pos];
3     s1[0] = s2[n+1] = 1;
4     for(int i = 1; i <= n; ++i) s1[i] = s1[i-1] * (pos - i) % p;
5     for(int i = n; i >= 1; --i) s2[i] = s2[i+1] * (pos - i) % p;
6     long long ans = 0;
7     for(int i = 1; i <= n; ++i)
8         up(ans, a[i] * s1[i-1] % p * s2[i+1] % p * inv[i-1] % p *
9             inv[n-i] % p * ((n-i) & 1 ? -1 : 1));
10    return ans;
11 }
```

韦达定理

2018年10月25日 19:10

一般地，在复数域内，设关于 x 的 n 次方程 $a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 = 0 (a_n \neq 0)$ 的 n 个根是 $x_i (1 \leq i \leq n)$ ，则有韦达定理（根与系数关系）如下：

$$\left\{ \begin{array}{l} \sum_{i=1}^n x_i = -\frac{a_{n-1}}{a_n} \\ \sum_{1 \leq i < j \leq n} x_i x_j = \frac{a_{n-2}}{a_n} \\ \cdots \cdots \cdots \\ \sum_{1 \leq i_1 < i_2 < \cdots < i_k \leq n} x_{i_1} x_{i_2} \cdots x_{i_k} = (-1)^k \frac{a_{n-k}}{a_n} \\ \cdots \cdots \cdots \\ \prod_{i=1}^n x_i = (-1)^n \frac{a_0}{a_n} \end{array} \right. \quad (3)$$

韦达定理的逆定理也成立，即：若 $x_i (1 \leq i \leq n)$ 满足（3）式，则 $x_i (1 \leq i \leq n)$ 一定是关于 x 的方程 $a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 = 0 (a_n \neq 0)$ 的 n 个根。

特别地，设一元三次方程 $ax^3 + bx^2 + cx + d = 0 (a \neq 0)$ 的三个根分别为 x_1, x_2, x_3 ，则有：

$$\left\{ \begin{array}{l} x_1 + x_2 + x_3 = -\frac{b}{a}, \\ x_1 x_2 + x_2 x_3 + x_3 x_1 = \frac{c}{a}, \\ x_1 x_2 x_3 = -\frac{d}{a}, \end{array} \right. \quad (4)$$

欧拉几何定理

2018年10月25日 11:21

【欧拉几何定理】

内容

1) 设三角形的外接圆半径为 R ，内切圆半径为 r ，外心与内心的距离为 d ，则 $d^2 = R^2 - 2Rr$ 。

2) 三角形 ABC 的垂心 H ，九点圆圆心 V ，重心 G ，外心 O 共线，称为 欧拉线

【欧拉拓扑定理】

$V + F - E = X(P)$ ， V 是多面体 P 的顶点个数， F 是多面体 P 的面数， E 是多面体 P 的棱的条数， $X(P)$ 是多面体 P 的欧拉示性数。

如果 P 可以同胚于一个球面（可以通俗地理解为能吹胀成一个球面），那么 $X(P) = 2$ ，如果 P 同胚于一个接有 h 个环柄的球面，那么 $X(P) = 2 - 2h$ 。

$X(P)$ 叫做 P 的拓扑不变量，是拓扑学研究的范围。

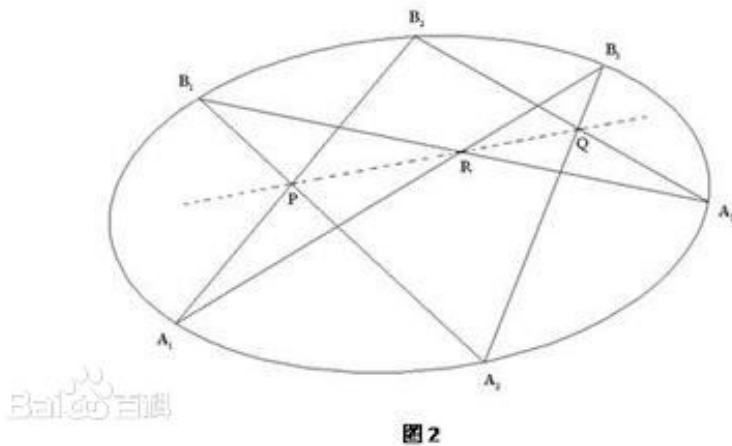
平面几何定理

2018年10月25日 15:55

【帕斯卡定理】

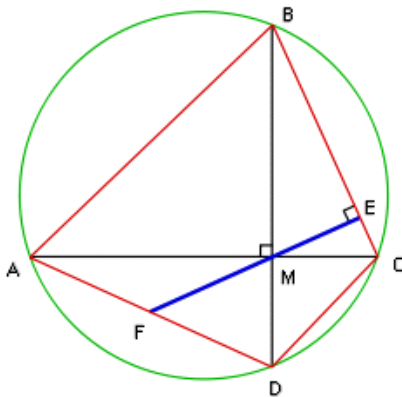
如果一个六边形内接于一条二次曲线(圆、椭圆、双曲线、抛物线)，那么它的三对对边的交点在同一条直线上。

在一条圆锥曲线上任取六点 A_1, A_2, A_3 和 B_1, B_2, B_3 ，设 A_1B_2 和 A_2B_1 交于点 P
 A_2B_3 和 A_3B_2 交于点 Q ， A_3B_1 和 A_1B_3 交于点 R ，则 P, Q, R 三点共线。(图2)



【婆罗摩笈多定理】

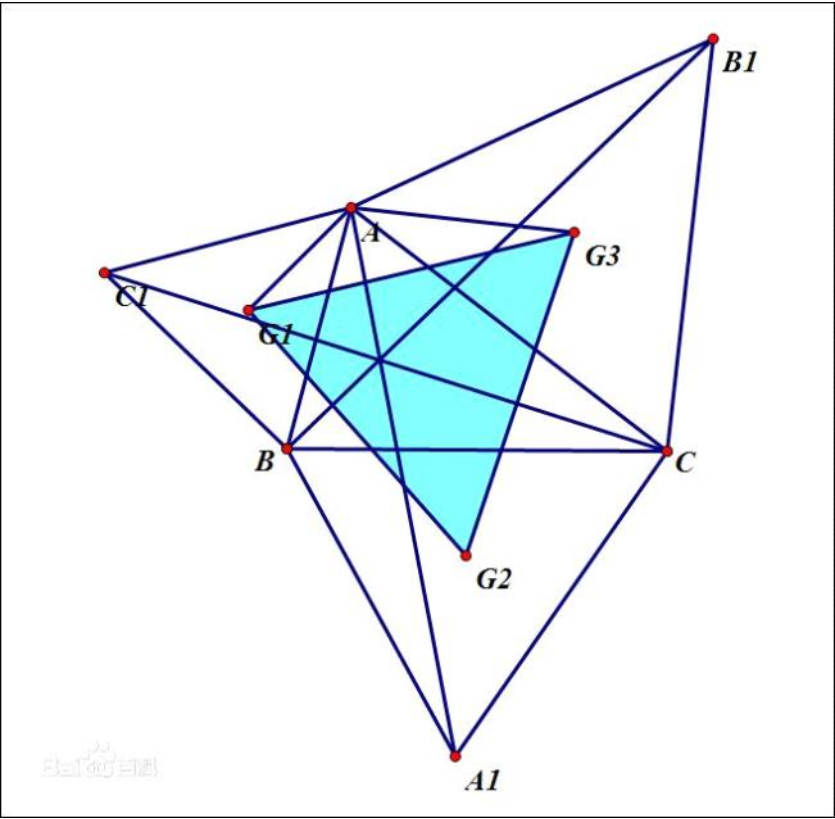
若圆内接四边形的对角线相互垂直，则垂直于一边且过对角线交点的直线将平分对边。



【拿破仑定理】

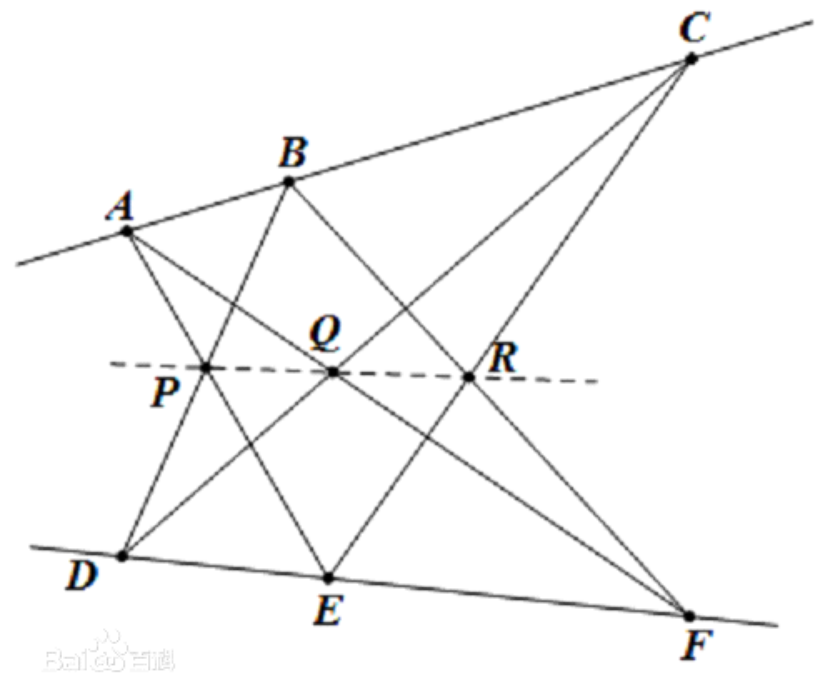
拿破仑定理则是法国著名的军事家拿破仑·波拿巴已知最早提出的一个几何定理：“以任意三角形的三条边为边，向外构造三个等边三角形，则这三个等边三角形的外接圆中心恰为另一个等边三角形的顶点。”该等边三角形称为拿破仑三角形。如果向内（原三角形不需为等边三角形）作三角

形，结论同样成立。



【帕普斯定理】

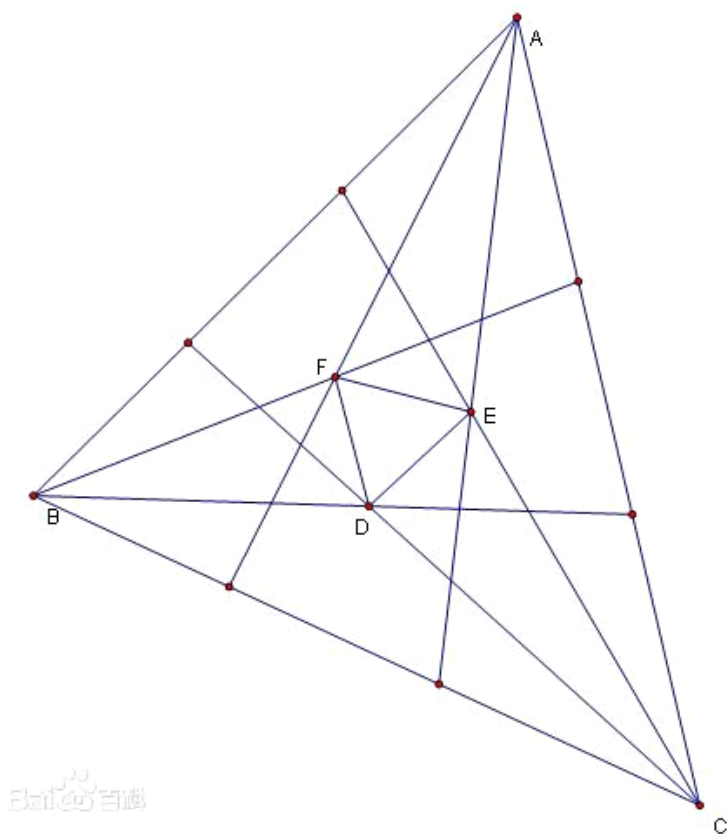
帕普斯 (Pappus) 定理，指的是直线 l_1 上依次有点A,B,C，直线 l_2 上依次有点D,E,F，设AE,BD交于P，AF,DC交于Q，BF,EC交于R，则P,Q,R共线。



【莫利定理】

莫利定理 (Morley's theorem)，也称为莫雷角三分线定理。将三角形的

三个内角三等分，靠近某边的两条三分角线相交得到一个交点，则这样的三个交点可以构成一个正三角形。这个三角形常被称作莫利正三角形。



【披萨定理】

披萨定理是平面几何学中的一个定理。它指出，如果以圆盘中任意一个指定点为中心，切下 n 刀，使相邻的两刀隔的角度相同；然后按顺时针（或逆时针）的顺序给切出的各块交替染上两种颜色，将圆盘分为两个部分。那么有下列结论：

当 n 是大于2的偶数（ $n=4, 6, 8, 10, 12, 14, \dots$ ），或有任一刀通过圆心时：两种颜色的部分面积一样大。

若任意一刀都不通过圆心，那么：

当 $n=1, 2$ 或 n 除以4余3（ $n=1, 2, 3, 7, 11, 15, \dots$ ）的时候，包含圆心的部分面积比较大。

当 n 大于4且除以4余1（ $n=5, 9, 13, \dots$ ）的时候，包含圆心的部分面积比较小。

这个定理之所以被称为披萨定理，是因为其中分区圆盘的方式类似于分披萨的过程。这个定理可以说明，当两个人用以上的方法分披萨的时候，谁能拿到更多的披萨。

【梅涅劳斯定理】

当直线交 $\triangle ABC$ 三边所在直线 BC, AC, AB 于点 D, E, F 时,

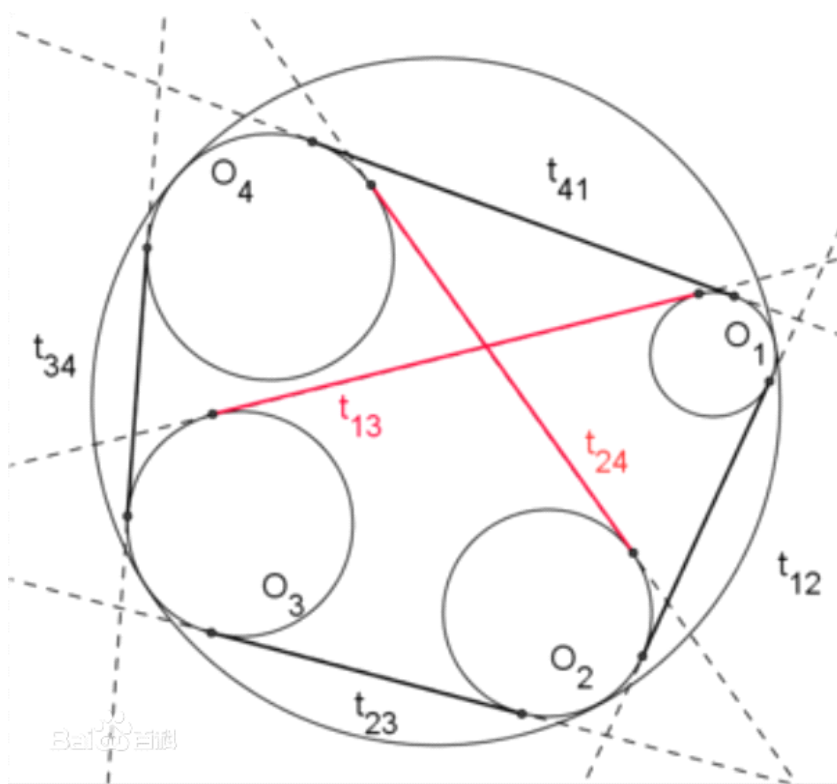
$$\frac{AF}{FB} \times \frac{BD}{DC} \times \frac{CE}{EA} = 1$$

【开世定理】

开世定理的背景是圆的内切圆。设有半径为 R 的一个圆 O ，圆内又有四个圆 O_1, O_2, O_3, O_4 内切于圆 O (如图1所示)。如果将圆 O_i, O_j 的外公切线的长度设为 t_{ij} ，那么开世定理声称，有下列等式成立。^[1]

$$t_{12} \cdot t_{34} + t_{14} \cdot t_{23} = t_{13} \cdot t_{24}$$

可以注意到，如果四个内切的圆都退化成点的话，就会变成圆 O 上的四个点，而开世定理中的等式也会化为托勒密定理。



【托勒密定理】

指圆内接凸四边形两对对边乘积的和等于两条对角线的乘积。

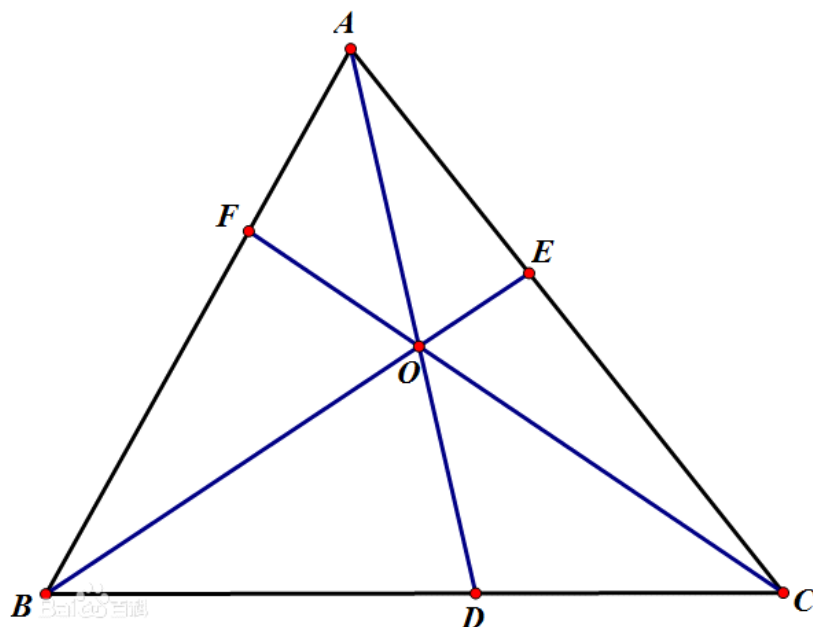
【托勒密不等式】

托勒密不等式：凸四边形的两组对边乘积和不小于其对角线的乘积，取等号当且仅当共圆或共线。

简单的证明：复数恒等式： $(a-b)(c-d) + (a-d)(b-c) = (a-c)(b-d)$ ，两边取模，得不等式 $AC \cdot BD \leq |(a-b)(c-d)| + |(b-c)(a-d)| = AB \cdot CD + BC \cdot AD$

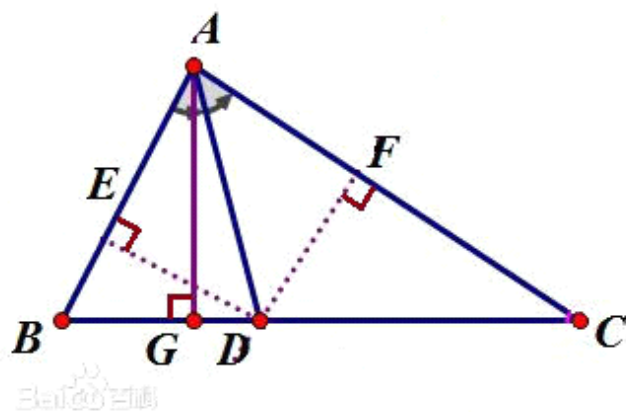
【塞瓦定理】

塞瓦定理是指在 $\triangle ABC$ 内任取一点 O ，延长 AO, BO, CO 分别交对边于 D, E, F ，则 $(BD/DC) \times (CE/EA) \times (AF/FB) = 1$ 。



【角平分线定理】

三角形一个角的平分线与其对边所成的两条线段与这个角的两边对应成比例。



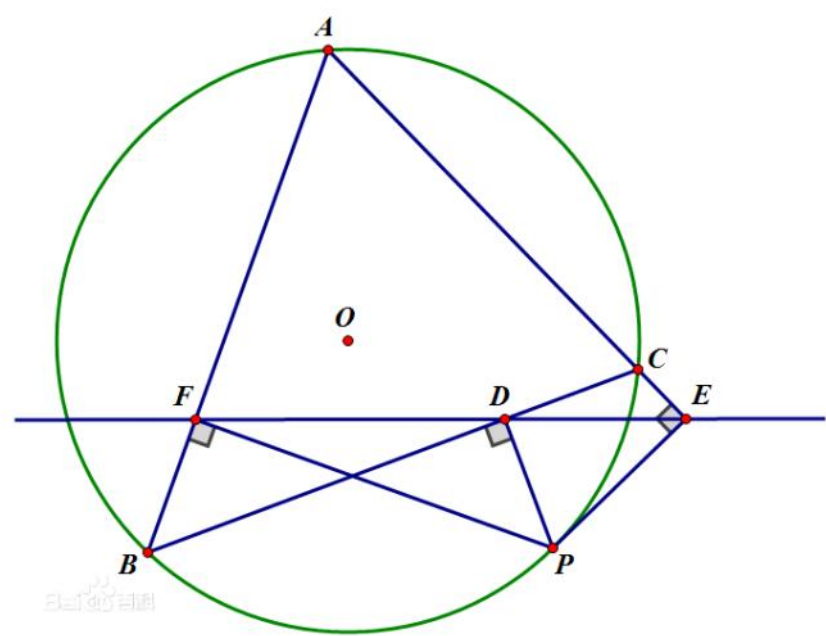
【西姆松定理】

西姆松定理是一个几何定理。表述为：过三角形外接圆上异于三角形顶点的任意一点作三边或其延长线上的垂线，则三垂足共线。（此线常称为西姆松线）。西姆松定理的逆定理为：若一点在三角形三边所在直线上的射影共线，则该点在此三角形的外接圆上。

相关的结果有：

- （1）称三角形的垂心为H。西姆松线和PH的交点为线段PH的中点，且这点在九点圆上。
- （2）两点的西姆松线的交角等于该两点的圆周角。
- （3）若两个三角形的外接圆相同，这外接圆上的一点P对应两者的西姆松线的交角，跟P的位置无关。
- （4）从一点向三角形的三边所引垂线的垂足共线的充要条件是该点落在

三角形的外接圆上。



格点几何定理

2018年10月25日 16:01

【皮克定理】

皮克定理是指一个计算点阵中顶点在格点上的多边形面积公式，该公式可以表示为 $2S=2a+b-2$ ，其中 a 表示多边形内部的点数， b 表示多边形边界上的点数， S 表示多边形的面积。

【闵可夫斯基定理】

坐标平面上任何包含原点的、面积大于4的、凸的、关于原点对称的闭区域一定含有异于原点的整点就是闵可夫斯基定理。

几何中的特殊点

2018年10月25日 16:08

【密克点】

命题 如图1,直线 AF 、 AE 、 BF 、 DE 两两相交成四个三角形,则这四个三角形的外接圆通过同一点 M .

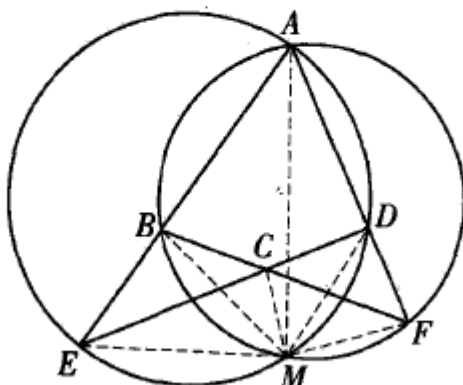


图1

【费马点】

“费马点”是指位于三角形内且到三角形三个顶点距离之和最短的点。

F为三角形ABC的费马点，O为任意一点，则：

$$\vec{OF} = \frac{\sin(A)}{\sin(A + \frac{\pi}{3})} \vec{OA} + \frac{\sin(B)}{\sin(B + \frac{\pi}{3})} \vec{OB} + \frac{\sin(C)}{\sin(C + \frac{\pi}{3})} \vec{OC}$$

反演几何学

2018年10月25日 16:50

请参阅资料：



利用反演变
换证明多...



反演变换的
应用

Szemerédi - Trotter theorem

2018年10月25日 22:18

Szemerédi - Trotter theorem

定理内容:

It asserts that given n points and m lines in the Euclidean plane, the number of incidences (i.e., the number of point-line pairs, such that the point lies on the line) is

$$O(n^{2/3} * m^{2/3} + n + m)$$

三维叉积的应用

2018年10月25日 22:19

一般方程法：直线的一般方程为 $F(x) = ax + by + c = 0$ 。既然我们已经知道直线的两个点，假设为 $(x_0, y_0), (x_1, y_1)$ ，那么可以得到 $a = y_0 - y_1, b = x_1 - x_0, c = x_0y_1 - x_1y_0$ 。

因此我们可以将两条直线分别表示为

$$F_0(x) = a_0x + b_0y + c_0 = 0, F_1(x) = a_1x + b_1y + c_1 = 0$$

那么两条直线的交点应该满足

$$a_0x + b_0y + c_0 = a_1x + b_1y + c_1$$

由此可推出

$$x = (b_0c_1 - b_1c_0)/D$$

$$y = (a_1c_0 - a_0c_1)/D$$

$$D = a_0b_1 - a_1b_0, (D \neq 0 \text{ 时, 表示两直线不平行})$$

向量除法

2018年10月25日 22:19

同时已知点乘和叉乘定义向量除法运算

若

$$\begin{cases} \vec{a} \cdot \vec{b} = c \\ \vec{a} \times \vec{b} = \vec{d} \end{cases}$$

,则有

$$\vec{a} \times (\vec{a} \times \vec{b}) = \vec{a}(\vec{a} \cdot \vec{b}) - \vec{b}(\vec{a} \cdot \vec{a})$$

解出

$$\vec{b} = \frac{\vec{a}(\vec{a} \cdot \vec{b}) - \vec{a} \times (\vec{a} \times \vec{b})}{\vec{a} \cdot \vec{a}} = \frac{\vec{a}c - \vec{a} \times \vec{d}}{\vec{a} \cdot \vec{a}}.$$

四、复数解释

当矢量为平面矢量时，其可与一复数对应。即

$$\begin{cases} \vec{a} = a_x \vec{i} + a_y \vec{j} \leftrightarrow a = a_x + ia_y \\ \vec{b} = b_x \vec{i} + b_y \vec{j} \leftrightarrow b = b_x + ib_y \end{cases}$$

则有

$$\bar{a}b = (\vec{a} \cdot \vec{b}) + i(\vec{a} \times \vec{b})$$

显然有

$$b = \frac{a(\vec{a} \cdot \vec{b}) + ia(\vec{a} \times \vec{b})}{|a|^2} (a \neq 0)$$

注意到其点积和叉积是福函数中的一对实部和虚部，若为一般解析函数即满足Cauchy条件。

相关性质和理解：

A: a和b的商的模等于a和b模的商

B: a和b的商所得向量的倾斜角等于a的倾斜角减去b的倾斜角。

C: 可以这样理解二维向量所蕴含的变换：

将某个二维向量所代表的平面点与原点连线的倾斜角加上一个幅角（逆时针），然后按照一定比例放大（乘法是乘模），除法是上述运算的逆

D: 从二元运算的角度，可以得到更进一步的理解：

假设a,b,c均为二维点或者二维的向量，令 $p=a/b, q=a-b$

可以这样理解除法和减法运算，他们表示的均为两个操作数的有序关系

也就是说，他们本身就存储了某些变换关系的具体参数，

可以具体的揭示出，利用某种或者某些变换，如何把第二个几何对象（右侧的）变换为第一个对象

所以：令方向向右的单位向量 $x=(1,0)$ ，原点为 $t=(0,0)$

则： $a/b=p/x, a-b=q-t$

他们都表示两个对象同时发生同种变换（参数相同），相对关系不变，均转换为某种单位对象和某个具体对象的关系，这个具体对象就是运算的结果；

除法蕴含旋转和伸缩，减法蕴含平移；

这个时候，如果用运算结果“右逆运算”某个向量： $c*p, c+p$ 就表示将这种关系，作用于新的对象

因此：乘法和加法是将一个关系作用于某一对象，而除法和减法是为了得到两个对象的具体关系（或者已知关系的具体参数）

事实上，平面上任两个等模的有向线段，均可以先通过旋转缩放，然后平移得到彼此 这样的变换是唯一的，参数的获取可以参见以下模板：

```
P r=(a[2]-a[1])/(b[2]-b[1]);  
P c=a[1]-b[1]*r;
```

注意：这里的旋转是指的是以原点为中心的旋转，如果旋转中心不是原点，可以先全体平移

上面代码中： r 是旋转伸缩向量, $r=|r|*(\cos t, \sin t)$,表示逆时针旋转 t ，然后伸缩 $|r|$ 比例

这样就得到了 b 线段是如何旋转伸缩到与 a 线段有向平行的线段的

$(b[1], b[2]) \xrightarrow{r} (b[1], b[2]) * r == (b[1] * r, b[2] * r)$

然后再看看 $(b[1] * r)$ 是如何平移得到 $a[1]$ 的，根据前面的理解就是代码中的 c 向量

事实上， (r, c) 已经是完备的关系表述了

因此： $y = x * r + c$ ，我们可以很轻松的把 x 点，通过相同的关系变换到目标点 y

简要模板：

```
P operator *(const P &a) const {
    return (P){x*a.x-y*a.y, x*a.y+y*a.x};
}

P operator /(const P &a) const {
    return (P){x*a.x+y*a.y, -x*a.y+y*a.x}/a.len()/a.len();
}
```