

初等数论三大定理和缩系乘法群

这篇和算法没什么关系，纯粹是基础知识。

初等数论三大定理，指将整个初等数论框架支撑起来的三个定理，分别是Fermat-Euler（费马欧拉）定理、Wilson（威尔逊）定理和Chinese-Residue（中国剩余）定理。

其中FE定理说明取模意义下缩系（简化剩余系/缩剩余系）集合的乘法构成群，Wilson定理揭示了模为素数的乘法群的结构，而CR定理阐述了怎样将群和群结合起来，即多素因子模数乘法群的结构问题。

它们三者的本质，都是解释缩系乘法群的结构问题。而研究缩系乘法群的结构，最终结论的形式是：奇素数幂次群结构、2的幂次群结构、CR定理，三个定理作为最终的最高结论。

Fermat-Euler定理

内容

设欧拉函数 $\varphi(n)$ 是0到 $n-1$ 里与 n 互素的数（缩剩余系）的个数，即缩系乘法群的阶。对于缩系中任一元素 a 有：

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

特别地，当 n 是单个素数 p 的时候 $\varphi(p)$ 是 $p-1$ 即：（费马小定理）

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

这其实是群论里的定理。任意一个群，群里任意一个元素，自乘群的阶次，一定会回到单位元。即：元素的阶整除群的阶。

证明也简单：对缩系所有元素同时进行乘法操作，构成缩系元素的一个置换。（也可以采用群论中陪集的方法）

这个定理在数学题或者算法中，一般用于简化幂次。例如快速幂函数。

推广

将研究对象转移到缩系以外。在完系（完全剩余系）中，任一元素 a 有相似结论：

$$a^{t+\varphi\left(\frac{n}{(a^t, n)}\right)} \equiv a^t \pmod{n}$$

对于足够大的整数 t 成立。意思是 a 本身自乘很多次后，也会落入循环中，循环节是 n 去除 a^t 与 n 最大公约数的缩系元素个数的约数。

并且这个足够大的 t 一般要求 a 与 n 重合的那部分素因数被“消除”干净了，即 a^t 这部分素因数的幂次已经达到或超过了 n 中的相应幂次。

这个证明是显然的，分素因数讨论即可。

由于欧拉函数的积性，循环节显然是 $\varphi(n)$ 的约数。因此弱化一下就是这样：

个素因子的缩系乘法群互不相干，分别构成相应的维度。

当已知这个数在各个维度的坐标，想求这个数的时候，利用线性代数的知识，先求各个维度上的单位向量，然后向量点乘即可。

单位向量的求法，就是一次不定方程。

模为奇素数幂的缩系乘法群的结构

构成循环群。生成元叫做原根。

不止这类模有原根，事实上1、2、4、奇素数的幂、2倍奇素数的幂都有，也就是说这些缩系乘法群也是循环群，而其余的模都没有。

模为2的幂的缩系乘法群的结构

是循环群与{-1,1}乘法群的笛卡尔积。

离散对数

这是一个天坑。关于离散对数的算法数不胜数，甚至是一个P与NP问题。如果未来的您能找到一个多项式时间求解离散对数问题的算法，那么今天的加密算法将半数失效，您不仅可以凭借这个算法轻松拿到图灵奖和菲尔兹奖，甚至可以改写世界历史。当然，如果您证明了不存在多项式时间的求解离散对数问题算法，相当于找到了P与NP问题的有效反例，照样可以拿到图灵奖和菲尔兹奖，只是无法改写历史的进程了而已。

由于本页面不打算涉及算法，那么这部分的算法计划将于暑假再开一个页面（这是因为烤漆实在没时间）。这里仅谈谈离散对数是怎么来的。

离散对数，就来源于循环群。我们知道，原根是缩系乘法群的生成元，那么每个元素是原根的多少次幂呢？

求解幂次，就是标准的对数运算。

我们知道，在复变函数里，指数函数是以 $2\pi i$ 为周期的，也就是说：

$$\ln re^{i\theta} = \ln r + i\theta + 2k\pi i \quad r > 0 \quad k \in \mathbb{Z}$$

这是因为 e 乘上 $2\pi i$ 就回到了乘法单位元1，和Fermat-Euler定理有着异曲同工之妙。

模n下，对于原根g如果g的t次方等于a那么有：

$$\log_g a \equiv t + k\varphi(n) \pmod n \quad k \in \mathbb{Z}$$

t只是对数的主值，即一个代表，一般取0到 $\varphi(n)$ （左闭右开）之间，以 $\varphi(n)$ 为周期。

例如模13的生成元是2，那么有表格：

n mod 13	1	2	4	8	3	6	12	11	9	5	10	7
$\log_2 n \pmod{13}$	0	1	2	3	4	5	6	7	8	9	10	11

更加神奇的是，如果引入取模下对数这个设定，那么换底公式是成立的，只要底一直是原根，并且除法的

