素数幂次问题

记号约定

在本文中，采用习惯记号，素数p在n中的幂次记为：

$$v_p(n)$$

代表p的这个幂次恰好整除n□而比这个值更高的幂次无法整除n□由于“ 恰好整除 ”记号（双竖线）容易和C语言“ 或 ”运算混淆，故不采用恰好整除记号。

另外一个记号也在后文出现□p进制下n的各位数字和：

$$S_p(n)$$

这个记号不一定要求p是素数，只是后文的p均为素数。

阅读本文时，希望能提前大致了解模p的缩系乘法群的相关知识。

p进赋值

写在前面

因为p进赋值的主体部分是数论一个艰深的分支，这里只阐述p进赋值的初始观点，不做深入研究，仅为了方便理解后文的内容。

p进赋值的基础，是p进制整数。这部分很简单，默认所有人都已经会了。

取模的观点

例如二进制数110110和101110。

如果我们模8的话，相当于取这两个数的最后三位，结果都为110。这也就是说，模8意义下无法区分两个数。

而模16的时候，相当于取这两个数的最后四位，分别为0110和1110。这个时候才能区分开两个数。

因此有这种感觉，模4就能区分的两个数，比模16才能区分的两个数，在二进赋值意义下距离更远。

p进赋值重新定义了两个数的距离。如果模p的a次幂才能够区分两个数，那么这两个数的距离就是1除以p的a次幂，即写成p进赋值表示后从右往左第几位才开始不同。因此□p进赋值表示的数无法像普通实数那样排在一条直线上，即没有通常意义的序关系。

因此，模数为素数幂次时，高幂次模数是对低幂次模数更加细化的划分。这样就理解了p进赋值中更靠左的位数更小，与一般的整数理解恰好相反。

也可以这样理解：多项式除法的时候，既可以使得余式的次数越除越高，也可以使得余式的次数越除越低。这依赖于除法的意义不同，导致除法的方向相反。

由于越靠左的位数表示距离越近，权重越小，因此p进赋值下的数左边可能会无限长，但右边一定有限长，即小数点后的位数有限。

p进赋值下的有理数

第一种解释，就是上述整数的除法。只是这次要求从右往左除。

类比一般的小数，这种解释也表明□p进赋值下的有理数，左边要么有限，要么无限循环。

因为有取模的意义，采用p进赋值的时候，有理数本身和它的表示一定是一一对应的，而不是像普通的p进制存在一二对应。

例如7进制下，0.6666666……和1表示同一个数，即每个有限小数都是一二对应，存在两种表示方法，而7进赋值下……6666666.0和1表示不同的数，至少在有理数范畴，实际的数与它的表示永远是一一对应的。

上面例子的……6666666.0事实上表示-1□p进赋值里没有负号。

第二种解释，采用数论倒数。

例如模13意义下，无法区分1/3和9，那么有理数1/3在13进赋值里最后一位就是9，并且是小数点前一位。如果想知道有理数1/3在13进赋值里倒数第二位是多少，就需要求解模169的时候3的倒数是多少，以此类推。

这两种解释完全等价。

采用p进赋值的时候，小数点后不为0的情况（合法p进赋值数必然有限长），就是有理数分母中存在p的情况。例如0.1表示1/p□左边无限循环的情况，就是分母中存在非p素因子的情况，也有可能代表负数。

采用p进赋值方法记录的数，加减乘除的规则都与普通的整数完全相同，只是除法方向相反而已。

因此可以找一个计数方法巧妙记录有理数，例如将1/3就记录成……10101010101等等，在数论方面的性质仍旧保留了。

p进赋值下的无理数

规定，每一个右端有限，左边无限不循环的合法p进赋值数定义一个p进赋值意义下的无理数。那么全体合法p进赋值数对应全体p进赋值意义下的实数。

p进赋值意义下的实数与正常的实数不同，往往不存在一一对应关系。一些正常的实数可能不对应于任何p进赋值意义下的实数，一些p进赋值意义下的实数也可能不表示任何正常实数。当然，也不保证两者一多对应的存在，因此事实上p进赋值创造了新的微积分体系。

例如熟悉的指对数计算，采用e^x和$\ln x$引入，计算时采用展开成无穷级数的方法。那么很多正常的实数就没办法表示，因为两个无穷级数在p进赋值里存在收敛域（收敛域不难计算），一个普通的x在正常实数范畴可以存在指对数，到p进赋值里就可能落在收敛域之外。又有可能在正常实数里负数无法计算对数（暂不考虑复数意义下），但是将负数表示为p进赋值之后可能落在收敛域里，就存在对数。

关于这部分的研究，已经是数学里很高深的部分。在此不再继续深入讨论。

升幂定理

总述

又叫LTE（Lifting The Exponent）现在统称升幂定理。本来是用于解决p进赋值问题的工具，但是由于它超好用而基础，已经深入到高中竞赛之中。

由于缩系乘法群的结构不同，根据素数为奇素数或2，分为LTEP定理和LTE2定理两部分。

LTEP

这是p为奇素数的情况。

使用本定理的前提：

第一条：p整除a-b，即模p意义下无法区分a和b；

第二条：p既不整除a，也不整除b，即a与b在p进制下末位不为0。

那么：a^n和b^n距离有多近呢？即究竟要取多大的模才能区分a^n和b^n呢？下面这个公式恒成立：

$$v_p(a^n-b^n)=v_p(a-b)+v_p(n)$$

即贡献分为两部分：原本的距离部分和指数的部分。

LTE2

这是p为2的情况。

使用本定理的前提：a和b都是奇数。

前提似乎与上面完全一样。由于群的结构不同，最终的结论也略有不同。事实上二进赋值与其他的p进赋值的性质不太一样。

$$v_2(a^n-b^n)=v_2(a-b)+v_2(a+b)+v_2(n)-1$$

贡献实际上也分为两部分：原本的距离部分和指数的部分，只是原本距离部分变复杂了。

素数在阶乘中的幂次

一般在解析数论研究中偏爱这个式子，最早是由Gauss研究的：

（一个无穷取整求和式，待补充）

这里推荐使用更加流行而简单的公式，替代上面这个繁杂的式子。它用到了文章开头的p进制下各位数字和：

$$v_p(n!)=\frac{n-S_p(n)}{p-1}$$

与等比数列求和公式很相似。由于涉及各位数字和，利用数学归纳法可以轻松证明。

素数在组合数中的幂次

（一个公式待补充）

如果仔细分析□p是否整除组合数其实和上下标在p进制下减法是否需要借位有关。这就有了下面的定理。

（待补充）

Lucas定理

结合上面“ 素数在组合数中的幂次 ”一同分析。上面的部分用于计算当组合数被p整除时，一共能被多少个p整除（仅判断模p的幂是否为0）；而这里则研究当组合数不被p整除时，模p余多少。

（待补充）

至于到算法层面，还有与中国剩余定理结合的扩展卢卡斯算法□exlucas□□用于解决模p的幂的余数问题。由于本文注重数学部分，这里不再讲解。