

# 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

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。

（待续）

## 升幂定理

### 总述

又叫LTE，这个英文缩写来源于高中数学竞赛以讹传讹的叫法，未知其确切含义。总之现在统称升幂定理。

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

（待续）

### LTE2

（待续）

## 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。

From: <https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link: <https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams.namespace:%E7%B4%A0%E6%95%B0%E5%B9%82%E6%AC%A1%E4%B8%8E%E8%BF%9B%E6%95%B0%E9%97%AE%E9%A2%98&rev=1591028862>

Last update: 2020/06/02 00:27