§1 整除性

1.1. 整除记号.

定义 1.1. 我们称 m 整除 n (或 n 可以被 m 整除), 当且仅当 $m>0$, 并且 $\frac{n}{m}$ 是一个整数. 可记作 $m\backslash n$. 即

$$m\backslash n \Leftrightarrow (m>0)\wedge(\exists k\in\mathbb{Z}, n=mk).$$

如果 m 不整除 n , 记作 $m \not\backslash n$.

定义 1.2 (最大公约数)**.** 两个整数 m,n 的最大公约数是最大的可以整除 m 和 n 的那个数. 即

$$\gcd(m,n)=\max\{k : k\backslash m \text{ 且} k\backslash n\}.$$

比如 $\gcd(12,18)=6$, 那么可以进行分数进行化简, 即 $\frac{12}{18}=\frac{12/6}{18/6}=\frac{2}{3}$.

注: 若 $n>0$, 那么 $\gcd(0,n)=n$, 因为任何一个整数都整除 0 .

定义 1.3 (最小公倍数)**.** 两个整数 m,n 的最小公倍数是最小的可以被 m,n 整除的那个数. 即

$$\operatorname{lcm}(m,n)=\min\{k>0; m\backslash k\wedge n\backslash k\}$$

1.2. 求最大公约数的算法.

算法描述: $\gcd(m,n)$ 是

- 若 $m=0$ 成立, 那么 $\gcd(m,n)=n$.
- 否则, $\gcd(m,n)=\gcd(n \bmod m, m)$.

这个递归式是有效的, 因为从公因式的角度来讲, 任何一个 m 和 n 的公因式都一定是 m 和 $n \bmod m$(即 $n-\lfloor\frac{n}{m}\rfloor m$) 的公因式. 下面只要考虑求出来的这个公因数一定是最大的即可.

可以构造一个方程, 来辅助说明这为什么是对的: 考虑找到 m',n', 满足 $m'm+n'n=\gcd(m,n)$, 其中, m',n' 由下面的内容给出:

- 如果 $m=0$, 那么置 $m'\leftarrow 0, n'\leftarrow 1$.
- 否则, 置 $r\leftarrow n \bmod m$, 并使用 r,m 代替 m,n(这是在倒着做上面的递归算法). 这时候我们需要知道他们前面的系数是什么. 因此需要找到 r,m 前面的系数, 记作 $\bar{r},\overline{m}$, 满足 $\bar{r}r+\overline{m}m=\gcd(r,m)$.
  - 由于 $r=n-\lfloor\frac{n}{m}\rfloor m$, 并且 $\gcd(r,m)=\gcd(m,n)$, 代入得

$$\bar{r}\left(n-\left\lfloor\frac{n}{m}\right\rfloor m\right)+\bar{m}m=\gcd(m,n).$$

  - 重写上述项, 收集 m,n 项, 就有了

$$\underbrace{\left(\bar{m}-\left\lfloor\frac{n}{m}\right\rfloor\bar{r}\right)}_{m'}m+\underset{n'}{\bar{r}}n=\gcd(m,n).$$

  - 因此我们可以将 $m'\leftarrow\bar{m}-\lfloor\frac{n}{m}\rfloor\bar{r}$, $n'\leftarrow\bar{r}$.

对于上述的过程, 给出一个例子, 如在求 $\gcd(12,18)$ 的时候表示出 6:

$$\begin{aligned} 6 &= 0.0 + 1 \cdot 6 \\ &= (1-0) \cdot 6 + 0.12 \\ &= (-1) \cdot 12 + 1 \cdot 18 \end{aligned}$$

接下来解释为什么是最大的: 假设上述算法得到 $\gcd(m,n) = d$, 以及 $m'm + n'n = d$, 但是存在一个 $d' > d$, 满足 $d \backslash m \wedge d \backslash n$. 由于任何 m, n 的公因子都要整除 $m'm + n'n$, 它也要整除 d, 因此它一定要小于 d. 这就达到了矛盾.

下面介绍几个常用的性质.

性质 1.1. $k \backslash m \wedge k \backslash n \Leftrightarrow k \backslash \gcd(m,n)$.

证明. 若 $k \backslash m, k \backslash n, k \backslash (m'm + n'n) \Rightarrow k \backslash \gcd(m,n)$. □

性质 1.2 (gcd 与公因子的关系)**.** 每一个公因子是他们最大公因数的因子

1.3. 对因子的求和及其常见变形.

性质 1.3 (因子的对偶性)**.**

$$\sum_{m \backslash n} a_m = \sum_{m \backslash n} a_{n/m} \quad , n > 0, n \in \mathbb{Z}.$$

性质 1.4 (求和记号的交换)**.**

$$\sum_{m \backslash n} \sum_{k \backslash m} a_{k,m} = \sum_{k \backslash n} \sum_{l \backslash (m/k)} a_{k,kl}$$

证明. 这是因为可以将求和记号的整除用 Iverson 括号来表达. 即

$$\sum_{m|n} a_m = \sum_{k} a_m [n = mk]$$

从而, 如果把等式的左手边记作 LHS, 右手边记作 RHS, 有

$$\begin{aligned} \text{LHS} &= \sum_{j,l} \sum_{k,m>0} a_{k,m} [n = jm][m = kl] = \sum_{j} \sum_{k,l>0} a_{k,kl} [n = jkl] \\ \text{RHS} &= \sum_{j,m} \sum_{k,l>0} a_{k,kl} [n = jk] \left[\frac{n}{k} = ml\right] = \sum_{m} \sum_{k,l>0} a_{k,kl} [n = mlk] \end{aligned}$$

从而等式的左手边等于等式的右手边. □

这同样可以使用直观的方式: 把要求的和式写成三角形, 然后从两个不同的方面进行描述即可.

§2 质数

下文中, 若无特殊说明, 总是用 p 代表一个质数.

2.1. 基本的定义.

定义 2.1 (质数)**.** 如果一个数 p 有且仅有两个正因数, 也就是 1 和 p, 我们称它为质数. 特别地, 1 不是质数.

定义 2.2 (合数)**.** 有大于 2 个正因子的数叫做合数. 特别地, 1 不是合数.

实际上, 大于 2 的数之间, 一个数要么是质数, 要么是合数. 但不可能同时是两者.

实际上, 质数是构成自然数的骨架. 我们给出如下的定理:

定理 2.1 (唯一分解定理)**.** 任何一个数 n 都可以分解为若干个质数的乘积.

$$n = p_1 \cdots p_m = \prod_{k=1}^{m} p_k, \quad p_1 \leqslant \cdots \leqslant p_m.$$

其中, $p_1, p_2, \cdots, p_m$ 都是质数. 并且这个分解式唯一.

存在性. 使用数学归纳法.

[唯一性] 考虑反证法. 假设同一个数存在两个不同的分解方法

$$\begin{aligned} n &= p_1 \cdots p_m \quad p_1 \leq p_2 \leq ... \leq p_m \\ &= q_1 \cdots q_k \qquad q_1 \leq q_2 \leq ... \leq q_k \end{aligned}$$

以及 p_i, q_i 都是质数.

$p_1 = q_1$ 假设 $p_1 < q_1$, 根据他们都是质数, 故 $\gcd(p_1, q_1) = 1$. 根据 Euclid 算法, 存在 a, b, 使得

$$ap_1 q_2 \cdots q_k + bq_1 q_i \cdots q_k = q_2 \cdots q_k.$$

$$\text{现在} p_1 \,|ap_1 q_2 \cdots q_k, p_1|\, bq_1 q_2 \cdots q_k, \Rightarrow p_1 \mid q_2 \cdots q_k$$

而这就意味着 $\frac{q_2 \cdots q_k}{p_1} \in \mathbb{Z}$, 且 $q_2 \cdots q_k$ 有一分解使 q_1 出现. 但是 $q_2 ... q_n < n$, 根据归纳法, 其必有唯一分解. 矛盾! 因此, $p_1 = q_1$. 此时可以在等两端同时除以 p_1, q_1. 重复刚才的过程, 可以证明 $p_2 = q_2$ 等.

□

注: 此式子有时候也可以记作

$$n = \prod_{p} p^{n_p}, \quad n_p \geqslant 0.$$

其中 $n_p \geq 0$, n_p 是质数 p 出现的次数.

由此我们便可以得到整数类似于 “坐标” 的表示. 只不过这里的维数是无穷维的.

例如, 下面方框里面的就可以认为是整数的 “坐标”.

$$12 = 2^{\boxed{2}} \times 3^{\boxed{1}} \times 5^{\boxed{0}} \times 7^{\boxed{0}} \times \cdots$$
$$18 = 2^{\boxed{1}} \times 3^{\boxed{2}} \times 5^{\boxed{0}} \times 7^{\boxed{0}} \times \cdots$$

从而, 12 和 18 的坐标可以记作

$$12 = \langle 2, 1, 0, 0\rangle, 18 = \langle 1, 2, 0, 0, \cdots\rangle$$

在这个坐标系下, 对于一个数的坐标形式, 即 $\langle n_2, n_3, n_5, n_7, \cdots\rangle$, 我们发现

$$\begin{aligned}
&k = mn && \Leftrightarrow k_p = m_p + n_p, && \forall p.\\
&m\backslash n && \Leftrightarrow m_p \le n_p \quad , && \forall p\\
&k = \gcd(m, n) && \Leftrightarrow k_p = \min(m_p, n_p), && \forall p\\
&k = \operatorname{lcm}(m, n) && \Leftrightarrow k_p = \max(m_p, n_p), && \forall p.
\end{aligned}$$

接下来看几个质数的性质.

性质 2.2. 质数有无穷多个.

证明. 假设有有限个个数的指数 $p_1, p_2, ..., p_k$, 总可以构造 $p_1p_2...p_k + 1$ 为一新的指数, 且无限的重复下去. □

性质 2.3. 第 n 个指数大约是 $n \ln n$.

这性质在这里无法证明. 需要的知识太多了. 但是我们可以用它来做复杂度的小估计.

2.2. 质数筛法.

a) Eratosthenes 筛 这种筛法采用如下的两步找到 $2 \sim x$ 的质数:

1. 写下 $[2..x]$ 的所有质数, 手指向 2.
2. 如果指向的元素没有被叉掉, 把它作为质数.
   - 将手指向的那个数的所有倍数叉掉
   - 手指移动到下一格, 如果移出了范围, 结束, 否则回到 2.

2.3. Euler 筛. 我们发现上述的 Esatosthenes 筛会把一个数重复筛掉多次. 如果我们让每个数都用其最小的质因数筛去, 就可以提升效率. 由于要用最小的质因数, 合数也要参与到筛的过程中.

要达到这种效果, 更好的办法是内层循环枚举在用第几个质数而非对这个质数乘几倍. 我们可以采用如下的办法:

1. 首先列出 $[2..n]$ 的所有数

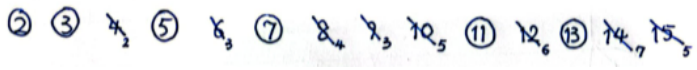
2. 把第一个数从列表里面拿出来, 创建一个新列表, 列表里面的内容是列表里面所有数 (包括第一个) 乘上刚刚拿出来的第一个数.
3. 把新列表里面出现的数从原列表移除
4. 输出列表里面的第一个数 (是一个质数) 并把它移除; 重复 $2 \sim 4$ 直到列表耗尽.

例如, 从列表 2 3 4 5 ... 30 开始. 然后新的列表是 4 6 8 10 ... 60. 减去前一个列表得到 2 3 5 7 9 ... 29. 现在 2 是质数, 然后对列表 3 5 7 9 ... 29 重复该过程. 下一步, 新的列表是 9 15 21 27 ... 87, 减去前一个列表得到 3 5 7 11 13 ... 29, 现在 2 和 3 是质数, 然后对列表 5 7 11 13 ... 29 重复该过程. 同样地, 对质数 5 和 7 进行处理, 并且由于 $7 \times 7 > 30$, 该过程停止, 剩下的列表是 11 13 17 19 23 29, 因此小于 30 的完整质数列表是 2 3 5 7 11 13 17 19 23 29.

但是这并不方便我们程序的书写. 与其把新列表创建出来之后删除其中的元素, 不如像刚刚那样划线. 我们给出如下的算法:

```
vector<int> primes;
bool is_prime[NR_PRIMES];
for(i=2..n){
    if(!is_prime[i]) primes.push_back(i);
    for(int p:primes){
        if(p*i>n) break;
        is_prime[p*i] = 1;
        if(i%p==0) break; // 再大就不是最小质因数了.
    }
}
```

例如前 15 个数的筛去过程如下:



§3 互素

3.1. 基本定义.

定义 3.1. 对于 $m, n \in \mathbb{Z}$, 如果 $\gcd(m, n) = 1$, 则称 m, n 互素. 有时候记为 $m \perp n$. 即

$$m \perp n \Leftrightarrow m, n \in \mathbb{Z} \wedge \gcd(m, n) = 1.$$

而且根据以往的经验, 对于两个整数, 总是可以将不是互素的数转换为互素的数, 通过

$$\frac{m}{\gcd(m, n)} \perp \frac{n}{\gcd(m, n)}$$

.

实际上,

$$\begin{aligned} m \perp n &\Leftrightarrow \min(m_p, n_p) = 0, \forall p \\ &\Leftrightarrow m_p n_p = 0 \quad , \forall p \end{aligned}$$

下面来看互素的数的性质.

性质 3.1. 若 $k \perp m$ 且 $k \perp n$, 则 $k \perp mn$.

证明. 这是因为 $\forall p, k_p m_p = 0, \quad k_p n_p = 0, k_p (m_p + n_p) = 0.$ □

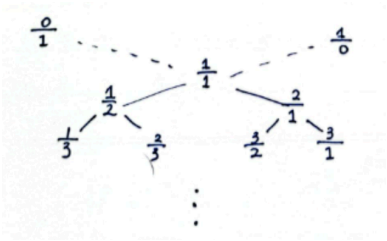
3.2. Stern-Brocot 树. 下面介绍一种方法, 可以构造出所有的分数 m/n. 并且可以通过这个构造表明有理数 $\mathbb{Q}$ 是可数的, 与实数 $\mathbb{R}$ 有本质不同.

1. 以 $\left(\frac{0}{1}, \frac{1}{0}\right)$ 开始, 然后一直重复一下操作:
   - 在两个分数 $\frac{m}{n}$ 与 $\frac{m'}{n'}$ 之间插入中间数 $\frac{m+m'}{n+n'}$.

例子 3.1.

$$(1)\frac{0}{1}, \boxed{\frac{1}{1}}, \frac{1}{0}$$
$$(2)\frac{0}{1}, \boxed{\frac{1}{2}}, \frac{1}{1}, \boxed{\frac{2}{1}}, \frac{1}{0}$$
$$(3)\frac{0}{1}, \boxed{\frac{1}{3}}, \frac{1}{2}, \boxed{\frac{2}{3}}, \frac{1}{1}, \boxed{\frac{3}{2}}, \frac{2}{1}, \boxed{\frac{3}{1}}, \frac{1}{0}.$$

这可以用树状的结构表示:



定理 3.2. Stern-Brocot 树构造了所有的既约分数. 而且每个数都仅仅出现了一次.

每个数仅仅出现了一次. 假设 m/n 和 m'/n' 是构造过程中出现的两个相同的分数. 那么有如下的式子成立:

$$m'n - mn' = 1.$$

使用归纳法,

- 初始的时候, 有 $1 \cdot 1 - 0 \cdot 0 = 1$.
- 当插入 $(m+m')/(n+n')$, 就变为

$$(m+m')\,n - m\,(n+n') = 1$$
$$m'\,(n+n') - (n+m')\,n = 1$$

, 化简后得到上面的式子. 因而此性质成立. 并且可以验证, 顺序关系 $\frac{m}{n} < \frac{m+m'}{n+n'} < \frac{m'}{n'}$ 总是成立. 进而不可能在两个不同的地方得到相同的数.

[没有数被漏掉] 假设 (a/b) 是一个不存在的数. 而且我们说

$$\frac{m}{n} = \frac{0}{1} < \underbrace{\left(\frac{a}{b}\right)}_{\text{不存在}} < \frac{1}{0} = \frac{m'}{n}.$$

在构造的某一阶段, 若 $\frac{m}{n} < \left(\frac{a}{b}\right) < \frac{m'}{n'}$, 有三种情况:

1. $\frac{m+m'}{n+n'} = \frac{a}{b}$, 说明这数存在, 矛盾!

2. $\frac{m+m'}{n+n'} < \frac{a}{b}$, 可以置 $m \leftarrow m+m', n \leftarrow n+n'$.
3. $\frac{m+m'}{n+n'} > \frac{a}{b}$, 可以置 $m' \leftarrow m+m', n' \leftarrow n+n'$.

此过程必定有限, 因为

$$\frac{a}{b} - \frac{m}{n} > 0 \wedge \frac{m'}{n'} - \frac{a}{b} > 0,$$

这就意味着 $an - bm \geqslant 1, \quad bm' - an' \geqslant 1$. 因此 $(m'+n')(an-bm) + (m+n)(bm'-an') \geqslant m'+n'+m+n$. 我们必定在 $a+b$ 步内结束这个算法. □

3.3. 由 Stern-Brocot 树构成的级数. 我们用 $\mathcal{F}_N$ 表示分母比 N 小的 $0 \sim 1$ 之间的既约有理数. 比如,

$$\mathcal{F}_6 = \frac{0}{1}, \frac{1}{6}, \frac{1}{5}, \frac{1}{4}, \frac{1}{3}, \frac{2}{5}, \frac{1}{2}, \frac{3}{5}, \frac{2}{3}, \frac{3}{4}, \frac{4}{5}, \frac{5}{6}, \frac{1}{1}$$

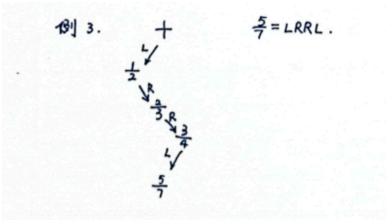
如何生成这一串数列? 实际上, 只要从 $\frac{0}{1}, \frac{1}{0}$ 开始, 按照 Stern-Brocot 树那样插入, 只要分母不是很大, 就一直可以插入. 多亏了它是不重不漏的.

实际上, 这一系列数列还给了我们一些观察:

1. 可以由 $\mathcal{F}_{N-1}$ 生成 $\mathcal{F}_N$. 只要在两个数之间按照上面构造法在每两个数中间插入对应的数即可.
2. 若 N 是质数, 有 $N-1$ 个数会出现; 否则, 新出现的数小于 $N-1$ 个.
3. $\mathcal{F}_N$ 给出了定理 "扩展 Euclid 算法" 的另一个证明: 因为我们可以让 $\frac{b}{a}$ 为 $\mathcal{F}_N$ 在 $\frac{m}{n}$ 之前的那个数, 进而得到 $ma - nb = 1$. 例如, $3a - 7b = 1, \quad \exists a = 5, b = 2$ 满足, 并且在 $\mathcal{F}_7$ 中 2/5 在 3/7 之前恰好 1 位.

3.4. Stern-Brocot 树作为有理数的表示. Stern-Brocot 树可以作为有理数的表示. 例如, 我们用 L 表示当前节点向左走, R 表示当前节点向右走. 特别地, 1 由空串表示.

例子 3.2. 例如, 5/7 可以用 LRRL 表示.



a) 给出 L 和 R 表示的序列, 求对应的有理数 由此我们便可以问: 给一个 L 和 R 的序列, 与之对应的有理数是什么?

首先定义

$$f(S) := \text{ 有理分数对应于} L5R \text{ 的序列} S$$

如 $f(LRRL) = \frac{5}{7}$. 由于分子和分母仅仅是前后两个的分子和分母的线性组合, 因此可以用 2×2 的矩阵表达.

从而定义

$$M(S) = \begin{pmatrix} n & n' \\ m & m' \end{pmatrix}.$$

那么

$$M(SL) = \begin{pmatrix} n & n+n' \\ m & m+m' \end{pmatrix} = \begin{pmatrix} n & n' \\ m & m' \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} = M(s) \underbrace{\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}}_{L}$$

$$M(SR) = \begin{pmatrix} n+n' & n' \\ m+m' & m' \end{pmatrix} = \begin{pmatrix} n & n' \\ m & m' \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} = M(s) \underbrace{\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}}_{R}$$

特别地, $M(I) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$. 所以, 只要做矩阵乘法, 就可以得到这一序列代表的有理数. 如

$$M(LRRL) = LRRL = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 3 & 4 \\ 2 & 3 \end{pmatrix}$$

我们最终的结果就是 $f(s) = f\left(\begin{pmatrix} n & n' \\ m & m' \end{pmatrix}\right) = \frac{m+m\prime}{n+n'}$..

b) 给出有理数, 求其对应的 L, R 序列 由于有理数的构造与 2×2 的矩阵乘法有对应, 根据大小关系干脆在 Stern-Brocot 树上面 "二分查找".

$S := 1$
while $\frac{m}{n} \neq f(s)$:
  if $\frac{m}{n} < f(S)$ then (output(L); $S \leftarrow SL$)
  else (output(R) : $S \leftarrow SR$).

下面给出另一种证明方法, 我们可以改变 m, n 的值, 而不是修改 S:

证明. 注意到

$$f(RS) = f\left(\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} n & n' \\ m & m' \end{pmatrix}\right) = f\left(\begin{pmatrix} n & n' \\ m+n & m'+n' \end{pmatrix}\right) = f(S) + 1$$

因而, 如果在 $m/n, m > n$ 上二分搜索的时候, 会首先输出 R. 同样的性质也对 L 有效, 即

$$\begin{aligned} \tfrac{m}{n} = f(RS) &\Leftrightarrow \tfrac{m-n}{n} = f(S), \quad m > n. \\ \tfrac{m}{n} = f(LS) &\Leftrightarrow \tfrac{m}{n-m} = f(S) \quad m < n. \end{aligned}$$

□

与之对应的算法是:

$$\begin{aligned}&\text{while } m<n:\\&\quad\text{if } m<n\text{then }\ \text{output}(L), n\leftarrow n-m\\&\qquad\text{else output}(R), m\leftarrow m-n\end{aligned}$$

例如, $m/n=\frac{5}{7}$, 就有

$m=5$		5	3	1	1
$n=7$		2	2	2	1
	L	R	R	R	

对于无理数而言, 这算法虽不能终止, 但可以用一个无穷序列来刻画.

§4 模运算. 同余关系

4.1. 同余记号. 借助 $a \bmod b$, 我们这次采用一个更加紧凑的符号来特别表示模某个数余数相同的那一类数:

定义 4.1. 我们记 $a \bmod m = b \bmod m$ 为

$$a\equiv b(\mathrm{mod}m).$$

读作: "在模 m 的意义下, a 与 b 同余 (具有相同的余数)".

在熟知上下文 (即 m 的值) 的时候, 可以省略后面的 $\mathrm{mod}m$.

这个符号和刚刚的使用 mod 的没有太大表意上的差别, 但是这记号的简洁性确启发了我们探索很多很有趣的性质. 下面我们来看这新的符号和原来的符号的联系.

性质 4.1.

$$a\equiv b(\mathrm{mod}m)\iff a-b \text{ 是 } m \text{ 的倍数}.$$

这样一来, 我们便像操作等式一样操作因数和倍数. 我们继续给出几个比较明显的性质:

性质 4.2. 在模 m 的意义下, 有

$$\begin{aligned}a\equiv b\wedge c\equiv d\quad&\Rightarrow\quad a+c\equiv b+d\\a\equiv b\wedge c\equiv d\quad&\Rightarrow\quad a-c\equiv b-d.\end{aligned}$$

性质 4.3. 在模 m 的意义下, 有

$$a\equiv b\wedge c\equiv d\quad\Rightarrow\quad ac\equiv bd$$

证明. 可以做拆分. 注意到 $ac - bd = (a - b)c + b(c - d)$, 是模数的倍数. □

我们发现模运算具有结合律, 我们可以把 k 个 a 相乘记作 a^k. 根据上面的结论, 我们就知道对于 $a, b \in \mathbb{Z}, n \geqslant 0$, 有 $a \equiv b \quad \Rightarrow \quad a^n \equiv b^n$.

但是在这样的体系下, 除法是空缺的. 比如 $3 \cdot 2 \equiv 5 \cdot 2 (\mathrm{mod} 4)$, 但是不能把两端的 2 约去得到 $3 \equiv 5$. 但是我们说明, 如果在某些条件下, 我们是可以把等号两边约去的 – 只要被约去的数和模数互素.

性质 4.4. 若 $a, b, d, m \in \mathbb{Z}, d \perp m$, 有

$$ad \equiv bd \Leftrightarrow a \equiv b (\mathrm{mod} m).$$

证明. 使用扩展 Euclid 算法, 可以找到 d', m', 使得

$$d'd + m'm = 1.$$

此时, 若 $ad \equiv bd$, 可以给两边乘上 d', 得到

$$add' = bd'd.$$

根据定义, 由于 $d'd \equiv 1$, 有 $ad'd \equiv a, bd'd \equiv b$, 从而得证. □

除了在等式两边除, 由于同余还有一个新的参数 – 模谁的意义下同余, 实际上还可以对这一参数变动.

性质 4.5.

$$ad \equiv bd (\mathrm{mod}\ md) \Leftrightarrow a = b (\mathrm{mod}\ m), d \neq 0$$

证明. 首先回忆 mod 具有分配率: $(a \bmod m)d = ad \bmod md$.

然后考虑

$$\begin{aligned} & a \bmod m = b \bmod m \\ \Leftrightarrow\ & (a \bmod m)d = (b \bmod m)d \\ \Leftrightarrow\ & ad \bmod md = bd \bmod md \Leftrightarrow ad \equiv bd (\mathrm{mod} md) \end{aligned}$$

□

结合上面的两个性质, 可以知道

推论 4.6. 对于 $a, b, m, d \in Z$, 有

$$\begin{aligned} & ad \equiv bd (\mathrm{mod} m) \\ \Leftrightarrow\ & a \equiv b \left(\mathrm{mod} \frac{m}{\gcd(d, m)} \right). \end{aligned}$$

这可以将除法那条性质中的右侧换为 $\gcd(d, m)$ 即可.

4.2. 模数的改变. 解同余方程组. 假若我们知道了 $a \equiv b(\mathrm{mod}100)$, 那么一定有 $a \equiv b(\mathrm{mod}10)$. 这就让我们想到

$$a \equiv b(\mathrm{mod}md) \Rightarrow a \equiv b(\mathrm{mod}m), d \in \mathbb{Z}$$

这是因为 md 的任意倍数都是 m 的倍数.

我们不妨将这个过程反过来. 若

$$\begin{aligned} a &= b(\mathrm{mod}m) \\ a &= b(\mathrm{\ mod\ }n) \end{aligned}$$

则一定有 $a \equiv b(\mathrm{mod}\,\mathrm{lcm}(m,n)), m, n > 0$.

从数的唯一分解定理可以得到这个结论.

假设我们希望找到一个数 x, 使得其满足

$$\begin{cases} x \equiv a_1 & \mathrm{mod}m_1 \\ x \equiv a_2 & \mathrm{mod}m_2 \\ x \equiv a_n & \vdots \mathrm{\ mod\ } m_n \end{cases}.$$

实际上, 我们可以故技重施, 先考虑简化版, 也就是只有两个方程的情况.

$$\begin{aligned} x &\equiv a_1 \mod m_1 \\ x &\equiv a_2 \mod m_2 \end{aligned}$$

把这个方程转化成普通的等号形式, $a_1 + t_1 m_1 = a_2 + t_2 m_2$, 可转化到解不定方程的时候的问题. 现在, 有 $\underbrace{t_1 m_1}_{xa} + \underbrace{(t_2(-m_2))}_{yb} = \underbrace{a_2 - a_1}_{c}$.

运用扩展 Euclid 算法, 就可得到这个 t_1 的一个特殊解答. 我们不妨记录作为 t_{11}.

由于特解转化为通解需要的是 $t_1 = t_{11} + k\,\underbrace{(-m_2)}_{b} / \gcd(\underbrace{m_1}_{a}, \underbrace{-m_2}_{b})$.

带入原式, 有 $x = a_1 + t_1 m_1 = a_1 + (t_{11} + k\,\underbrace{(-m_2)}_{b} / \gcd(\underbrace{m_1}_{a}, \underbrace{-m_2}_{b})) \times m_1$.

于是整理得到:

$$x = a_1 + t_{11} m_1 - k\,\mathrm{lcm}(m_1, m_2).$$

于是我们就可以下结论, 可以把两个方程合并为一个形如 $x \equiv a_{12} \mod \mathrm{lcm}(a_1, a_2)$. 只需要经过一些循环使得方程合并为一个即可.

§5 独立剩余系

5.1. 基本定义. 剩余系指的是给一系列互素的数, 将每一个数表示为分别模它们得到的结果的序列. 也就是说对 $\forall i,j,\quad 1\leqslant j<k<r,\quad m_j\perp m_k$,

$$\mathrm{Res}(x):=(x \bmod m_1, x \bmod m_2, \cdots, x \bmod m_r)\,.$$

这可以让我们知道关于 x 的性质.

例子 5.1. 考虑下面的 $x\in[0..14]$. 对于质数 3, 5; 以及他们的乘积 (15). 我们可以在行与行之间寻找规律:

$x \bmod 3$	$x \bmod 5$	$x \bmod 15$
0	0	0
1	1	1
2	2	2
0	3	3
1	4	4
2	0	5
0	1	6
1	2	7
2	3	8
0	4	9
1	0	10
2	1	11
0	2	12
1	3	13
2	4	14

上例子中在 15 以内的任意一对 $(x \bmod 3, x \bmod 5)$ 都是不同的. 可以看做每个数在 3 上和 5 上的 "分量". 我们甚至可以对分量做加减, 乘法. 如: $7=(1,2),\quad 13=(1,3)$, 那么 $7\times 13=(1\times 1,(2\times 3) \bmod 5)=(1,1)=1(\mathrm{mod}15)$.

从而, 便可以在某些情况下使用剩余系来表示整数. 首先来看大小为 2 的剩余系 (也是最简单的情况). 假设

$$(1,0)=a$$
$$(0,1)=b$$

那就可以推出我们要表示的数满足 $(x,y)=(ax+by) \bmod 15$.

对于上述的例子, 可以查上例表得到 $a=10,b=6$. 但是一般情况呢?

重新组织问题的表述: 对于 $m \perp n$, 如何找到满足

$$\begin{aligned} a \bmod m &= 1 \\ a \bmod n &= 0 \\ b \bmod m &= 0 \\ b \bmod n &= 1 \end{aligned}$$

的 a, b?

还是使用 Euclid 算法, 我们可以找到 m', n' 使得 $m'm + n'n = 1$. 取 $a = m'm, \quad b = n'n$ 即可.

例子 5.2. 有多少个 x 满足 $x^2 \equiv 1(\mathrm{mod}m)$? 我们认为 x 与 x' 是相等的, 当且仅当 $x \equiv x'(\mathrm{mod}m)$.

解答: 由于整数可以表示为若干个素数之积, 因此只要在分解式上归纳即可.

假设 m 是某个素数 p 的 k 次方, 即 p^k, 则可以重写为

$$(x-1)(x+1) \equiv 0 \pmod{p^k}.$$

1. 如果 $p > 2$,

$$p^k \backslash (x-1)(x+1) \Leftrightarrow \left(p^k \backslash (x-1)\right) \vee \left(p^k \backslash (x+1)\right),$$

有两个解.

2. $p = 2$, 若 $2^k \backslash (x-1)(x+1)$, 有
   (a) 要么 $x-1$, 要么 $x+1$ 可以被 2 整除, 但是不能被 4 整除; 另一个被 2^{k-1} 整除. ($k \geq 3$ 的时候后 4 个解, $\pm 1, 2^{k-1} \pm 1$). (例如: $p^k = 8$, 解为 1, 3, 5, 7; 以及每一个正奇数的平方可以表示为 $8n+1$ 的形式)

根据唯一分解定理, $x^2 \equiv 1(\mathrm{mod}m) \Leftrightarrow x^2 \equiv 1 \left(\mathrm{mod}p^{m_p}\right)$, $\forall p, m_p > 0$. 且 $\prod_{p=1}^{n} m_p = m$. 由于每一个质数独立于其他的质数, 除了 $p = 2$ 的情况, 他们可以互相组合.

因而总共有 2^r 个解, 如果不存在 $m_i = 2$ 的情形.

反而, 若包含 $m_i = 2$ 的情形, 共有 $2^{r+[8\backslash m]+[4\backslash m]-[2\backslash m]}$ 个解 (从上面的分析推出).

例如 $12 = 3 \times 2^2$, 有 $x^2 \equiv 1(\mathrm{mod}12) \Rightarrow x = 1, 5, 7, 11$; $15 = 3 \times 5$, 有 $x^2 \equiv 1 \pmod{15} \Rightarrow x = 1, 4, 11, 14$.

§6 幂次方的同余式

在第三章中, 我们希望证明:

性质 6.1.

$$0 \bmod m, n \bmod m, 2n \bmod m, (m-1)n \bmod m$$

仅仅由 d 份 $0, d, 2d, \cdots, m-d$ 这 $\frac{m}{d}$ 个数构成. 其中 $d = \gcd(m, n)$.

例子 6.1. $m=12\ n=8$ ， 那 $d=4$ ， 数列为 $0,8,4,0,8,4,0,8,4$.

证明. 会得到 d 份前 m/d 个值 这是因为

$$jn=mn(\mathrm{mod}m)\Leftrightarrow j\frac{n}{d}=m\frac{n}{d}\left(\mathrm{mod}\frac{m}{d}\right)$$

因此我们得到了当 $0\leq k<m/d$ 时这 d 个元素.

这 m/d 个元素恰好就是 $\{0,d,2d,\cdots,m-d\}$ 令 $m=m'd,\quad n=n'd$, 根据分配率, $kn \bmod m=d\,(kn' \bmod m')$. 所以在 $0\leq k<m'$ 的时候恰好会出现 d 次

$$0 \bmod m', n' \bmod m', 2n' \bmod m', \cdots, (m'-1) \bmod m'.$$

由于 $d=\gcd(m,n)$, 故 $m'\perp n'$. 所以只需要证上述的 m' 个值为 $\{0,1,...,m-1\}$. 由于这 m' 个数是互相不一样的, 且 $m'\perp n'$, 那么

$$jn'\equiv kx'\,(\mathrm{mod}m')\Leftrightarrow j\equiv k\,(\mathrm{mod}m')$$

. 并且根据鸽笼原理, 这 m 个数一定会填满 $0,1,...,m-1$.

□

性质 6.2 (Fermat 小定理)**.** 若 $n\perp p$, 则 $n^{p-1}\equiv 1(\mathrm{mod}p)$.

证明. 记 p 是一个质数. 由于有 $p-1$ 个数 $n \bmod p, 2n \bmod p, \cdots, (p-1)n \bmod p$ 是 $1,2,...,p-1$ 的某个排列, 因而把它们乘在一起有

$$\begin{aligned} n(2n)\cdots((p-1)n) &\equiv (n \bmod p)(2n \bmod p)\cdots((p-1) \bmod p) \\ &\equiv (p-1)! \\ &(\mathrm{mod}p) \end{aligned}$$

而这意味着 $(p-1)!n^{p-1}\equiv(p-1)!(\mathrm{mod}p)$. 约去 $(p-1)!$, 因为 $(p-1)\not| p$, 得到 $n^{p-1}\equiv 1(\mathrm{mod}p)$. □

例子 6.2. Fermat 曾怀疑 $2^{2^n}+1$ 生成一系列素数. 但是 $2^{32}+1$ 不是素数. 原因是, 置上述的定理的 $n=3$, 得到 $3^{2^{32}}\equiv 1\,(\mathrm{mod}2^{32}+1)$. 如果 $2^{32}+1$ 是质数的话. 但是, $3^{2^{32}}=3029026160\,(\mathrm{mod}2^{32}+1)$. 所以它不是质数.

性质 6.3. 对于 $n>1$ 的数,

$$(n-1)!\equiv -1(\mathrm{mod}n)\Leftrightarrow n \text{ 是质数}$$

证明. $\Leftarrow$: 若 $n>1$, n 不是质数, 会有一个质因子 p 出现在 $(n-1)!$ 的因子中, 从而 $(n-1)!\not\equiv -1$.

$\Rightarrow$: 考虑将一个数与它的逆配对 回顾, 若 $n\perp p$, 则存在 n', 满足 $n'n\equiv 1(\mathrm{mod}p)$. 由于 n 的任何两个逆元必须同余, 即 $nn'\equiv nn''\Rightarrow n'\equiv n'$. 如果我们都可以把它们成功配对, 那么 $(p-1)!$ 应该同余于-1.

若 $p=5, 4!=24(\mathrm{mod}5)$. 我们列出之后发现, 有些数自己是自己的逆元!

$$\underbrace{1^{-1}=1}_{\text{自己是自己的逆}} \quad 2^{-1}=3 \quad 3^{-1}=2 \quad 4^{-1}=4.$$

决定谁是自己的逆元 若 x 是自己的逆, $x^2\equiv 1(\mathrm{mod}p)$. 我们已经在上节证明如果 $p>2$ 恰有 2 个根, 1 和 $p-1$. 因此余下的可以配对, 得到 $(p-1)!\equiv 1\times(p-1)\equiv -1$.

当 $p=2$ 的时候, $(p-1)!\equiv -1$. □

值得说明的是, 这个仅仅是有理论价值, 而不是很好实际计算.

6.1. Euler 的 φ 函数. 小于等于 n 的与 n 互质的数有多少个? 欧拉的 φ 函数给出了解答.

定义 6.1. 定义 $\varphi(m):=\{0,1,\cdots,m-1\}$ 中与 m 互质的数的个数, 称为 Eulerφ 函数.

例子 6.3. 实际上,

$$\begin{aligned}&\varphi(1)=1,\\ &\varphi(p)=p-1 \qquad p \text{ 是质数},\\ &\varphi(m)<m-1 \quad m \text{ 是合数},\end{aligned}$$

引进了 Euler 函数, Euler 发现 Fermat 小定理可以推广为为 $n^{\varphi(m)}\equiv 1(\mathrm{mod}m)$. 对 $n\perp m$.

下面考虑 $\varphi(m)$ 的计算方法.

1. 如果 m 是某一质数 p 的 m 次方, 那么每隔 p 个就会划去一个. 即 $n\perp p^k\Leftrightarrow p\not| n$. 也就是说 $\{0,1,\cdots,p^k-1\}$ 中 p 的倍数有 $\{0,p,2p,\cdots,p^k-p\}$ 共计 p^{k-1} 个不与 p^m 互质的. 故 $\varphi\left(p^k\right)=p^k-p^{k-1}$. 例如 $\varphi\left(2^3\right)=4$. 这是因为可以表示为 ~~0~~ 1 ~~2~~ 3 ~~4~~ 5 ~~6~~ 7.
2. 如果 $m>1$, 且 $m=m_1m_2\,(m_1\perp m_2)$, 由上一节的剩余系的知识可以把 $0\le n<m$ 的数表示做唯一的对 $(n \bmod m_1, n \bmod m_2)$. 即

$$n\perp m\Leftrightarrow (n \bmod m_1)\perp m_1\wedge(n \bmod m_2)\perp m_2$$

那么这样的 "互素" 性质就可以被传递下去. 即计算 $\varphi\left(m_1\right)\varphi\left(m_2\right)$. 由于 m_1,m_2 互素, 没有公共的因子, 因此使用乘法原理就可以得到最终答案. 例如, $\varphi(12)=\varphi(3)\varphi(4)=2\times 2=4$.

上面的推导实际上蕴含了这一函数的特别好的性质: 积性.

定义 6.2 (积性函数)**.** 一个函数 $f(m)$ 称为积性函数, 如果 $f(1)=1$, 并且

$$f(m_1m_2)=f(m_1)f(m_2), \text{只要} m_1\perp m_2.$$

例子 6.4. φ 是积性函数; $x^2\equiv 1(\mathrm{mod}m)$ 的个数是积性的. $f(m)=m^d$ 是积性函数.

积性函数与唯一分解定理也联系密切. 某一函数具有积性意味着

$$f(m)=\prod_{p} f\left(p^{m_p}\right), 若 m=\prod_{p} p^{m_p}.$$

这样一来, 我们就得到了计算 φ 函数的方法:

$$\varphi(m)=\prod_{p\backslash m}\left(p^{m_p}-p^{m_p-1}\right)=m\prod_{p\backslash m}\left(1-\frac{1}{p}\right).$$

比如, $\varphi(12)=(4-2)(3-1)=12\left(1-\frac{1}{2}\right)\left(1-\frac{1}{3}\right)$.

性质 6.4.

$$\sum_{d\backslash m}\varphi(d)=m.$$

不妨以 12 做例子. 考虑所有以 12 为分母的可能未化为最简的真分数, 即

$$\frac{0}{12},\frac{1}{12},\frac{2}{12},\frac{3}{12},\frac{4}{12},\frac{5}{12},\frac{6}{12},\frac{7}{12},\frac{8}{12},\frac{9}{12},\frac{10}{12},\frac{11}{12}$$

将其化简, 就有

$$\frac{0}{1},\frac{1}{12},\frac{1}{6},\frac{1}{4},\frac{1}{3},\frac{5}{12},\frac{1}{2},\frac{7}{12},\frac{2}{3},\frac{3}{4},\frac{5}{6},\frac{11}{12}$$

按照分母分组, 就有 $\boxed{\frac{0}{1}}\boxed{\frac{1}{2}}\boxed{\frac{1}{3},\frac{2}{3}}\boxed{\frac{1}{4},\frac{3}{4}}\boxed{\frac{1}{6},\frac{5}{6}}\boxed{\frac{1}{12},\frac{5}{12},\frac{7}{12},\frac{11}{12}}$. 其中, 分母是 $\varphi(d)$, 一组里面有几个就是 $\varphi(d)$ 的值. 对于任意的数都是这样的. 这个性质是成立的.

我们看到, 可以将一个整数 m 分解为一个积性函数之和. 实际上, 分解前后, 这函数的积性保持不变.

性质 6.5. 如果 $g(m)$ 是一个积性函数, 且

$$g(m)=\sum_{d\backslash m} f(d).$$

说明 f 是积性函数.

证明. 对 m 施以归纳法. 首先看到, $f(1)=g(1)=1$.

- 当 $m>1$, 假设 $f\left(m_1m_2\right)=f\left(m_1\right)f\left(m_2\right)$, 只要 $m_1\perp m_2, m_1m_2<m$. 若 $m=m_1m_2,\quad m_1\perp m_2$, 有

$$\begin{aligned} g\left(m_1m_2\right)&=\sum_{d\backslash m_1m_2} f(d)=\sum_{d_1\backslash m_1}\sum_{d_2\backslash m_2} f\left(d_1d_2\right)\\ &\overset{IH}{=}\left(\sum_{d_1\backslash m_1} f\left(d_1\right)\sum_{d_2\backslash m_2} f\left(d_2\right)\right)-\underbrace{f\left(m_1\right)f\left(m_2\right)+f\left(m_1m_2\right)}_{当m_1m_2=m的时候}\\ &=g\left(m_1\right)g\left(m_2\right)-f\left(m_1\right)f\left(m_2\right)+f\left(m_1m_2\right). \end{aligned}$$

由于 $g\left(m_1m_2\right)=g\left(m_1\right)g\left(m_2\right)$, 因此 $f\left(m_1m_2\right)=f\left(m_1\right)f\left(m_2\right)$.

□

实际上, 上述的逆命题也是对的. 这就让我们可以对积性函数表达为另一个积性函数之和.

6.2. Mobius 函数: $\mu(n)$.

定义 6.3 (Mobius 函数). $\mu(m)$ 定义做

$$\sum_{d\backslash m}\mu(d)=[m=1], m\geqslant 1.$$

实际上这是个递归式. 我们刚刚没有给出其的表达. 这个函数在源于用两个积性函数互相表示的时候做的尝试. 下面来看这个性质.

性质 6.6.

$$g(m)=\sum_{d\backslash m}f(d)\quad\Leftrightarrow\quad f(m)=\sum_{d\backslash m}\mu(d)g\left(\frac{m}{d}\right).$$

证明. $\Leftarrow$:

$$\begin{aligned}\sum_{d|m}\mu(d)g\left(\frac{m}{d}\right)&\xlongequal{\text{共轭因子}}\sum_{d\backslash m}\mu\left(\frac{m}{d}\right)g(d)\\&=\sum_{d\backslash m}\mu\left(\frac{m}{d}\right)\sum_{k\backslash d}f(k)\\&=\sum_{k\backslash m}\sum_{d\backslash(\frac{m}{k})}\mu\left(\frac{m}{kd}\right)f(k)\\&=\sum_{k\backslash m}\sum_{d\backslash\left(\frac{m}{k}\right)}\mu(d)f(k)\\&=\sum_{k\backslash m}[m/k=1]f(k)=f(m)\end{aligned}$$

$\Rightarrow$: 对两边乘以 $\mu(m/d)$ 并求和:

$$\begin{aligned}\sum_{d\backslash m}g(d)\mu\left(\frac{m}{d}\right)&=\sum_{d\backslash m}\sum_{l\backslash d}f(l)\mu\left(\frac{m}{d}\right)\\&=\sum_{l\backslash m}\sum_{d\backslash l}f(l)\mu\left(\frac{m}{ld}\right)\\&=\sum_{l\backslash m}f(l)\sum_{d\backslash(\frac{m}{l})}\mu(d)\\&=\sum_{l\backslash m}f(l)=[m=l]=f(m).\end{aligned}$$

□

下面考虑如何计算 μ. 根据上述性质, μ 应该是积性函数. 因而只要知道 $\mu(p^k)$ 如何算即可.

如果 $m=p^k$, 根据 μ 的定义, $\mu(1)+\mu(p)+\mu\left(p^2\right)+\cdots+\mu\left(p^k\right)=0$, 对于任意 $k>1$, p^k 的因子为 $1, p, ..., p^k$. 又因为 μ 是积性函数, 所以 $\mu(1)=1$. 这表明, $\mu(p)=-1$, 对于任意的 $k>1$, $\mu(p^k)=0$.

所以根据以上的推导, 我们得到计算 μ 的方法:

性质 6.7.

$$\mu(m)=\prod_{p\backslash m}\mu\left(p^{m_p}\right)=\begin{cases}(-1)^r, & \text{若}m=p_1p_2\cdots p_r\\ 0, & \text{若}m\text{ 可被某个}p^2\text{ 整除.}\end{cases}$$

例子 6.5. 使用刚刚的定理对于

$$\sum_{d\backslash n}\varphi(d)=n$$

应用, 然后得到

$$\varphi(m)=\sum_{d\backslash m}\mu(d)\frac{m}{d}.$$

例如, $\varphi(12)=12-6-4+0+2+0=4$. 这与 $\varphi(m)=m\left(1-\frac{1}{p_1}\right)\cdots\left(1-\frac{1}{p_r}\right)$ 得到的结果相同.