§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 $n-\left\lfloor\frac{n}{m}\right\rfloor m$) 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 $\overline{r}, \overline{m}$, 满足 $\overline{r}r+\overline{m}m=\gcd(r,m)$.
  - 由于 $r=n-\left\lfloor\frac{n}{m}\right\rfloor m$, 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 $m' \leftarrow \bar{m}-\left\lfloor\frac{n}{m}\right\rfloor \bar{r}$, $n' \leftarrow \bar{r}$.

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".

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

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 $p_1 p_2 ... p_k + 1$ 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

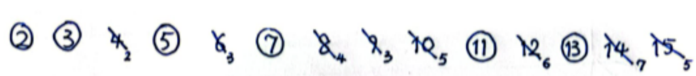
2. 把第一个数从列表里面拿出来, 创建一个新列表, 列表里面的内容是列表里面所有数 (包括第一个) 乘上刚刚拿出来的第一个数.
3. 把新列表里面出现的数从原列表移除
4. 输出列表里面的第一个数 (是一个质数) 并把它移除; 重复 2 ∼ 4 直到列表耗尽.

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

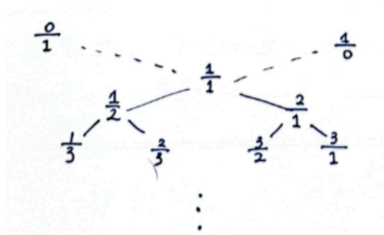
3.2. Stern-Brocot 树. 下面介绍一种方法, 可以构造出所有的分数 m/n. 并且可以通过这个构造表明有理数 $\mathbb{Q}$ 是可数的, 与实数 $\mathbb{R}$ 有本质不同.

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,

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

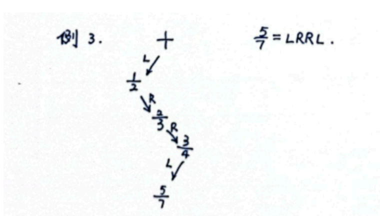
如何生成这一串数列? 实际上, 只要从 $\frac{0}{1}, \frac{1}{0}$ 开始, 按照 Stern-Brocot 树那样插入, 只要分母不是很大, 就一直可以插入. 多亏了它是不重不漏的.

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".

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

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

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

□

与之对应的算法是:

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

例如, $m/n = \frac{5}{7}$, 就有

$$
\begin{array}{ccccccccc}
m = 5 & & 5 & & 3 & & 1 & & 1 \\
n = 7 & & 2 & & 2 & & 2 & & 1 \\
& L & & R & & R & & R &
\end{array}
$$

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.