约数

李淳风

长郡中学

2024 年 8 月 1 日

定义

若整数 n 除以整数 d 的余数为 0，即 d 能整除 n，则称 d 是 n 的约数，n 是 d 的倍数，记为 $d|n$。

算数基本定理的推论

在算数基本定理中，正整数 n 被唯一分解为 $n = p_1^{c_1} p_2^{c_2} \cdots p_m^{c_m}$，其中 c_i 都是正整数，p_i 都是质数，且满足 $p_1 < p_2 < \cdots < p_m$，则 n 的正约数集合可以写作：

$$\{p_1^{b_1} p_2^{b_2} \dots p_m^{b_m}\}, \text{其中 } 0 \le b_i \le c_i$$

n 的正约数个数为 ($\prod$ 表示连乘)：

$$(c_1 + 1) * (c_2 + 1) * \cdots * (c_m + 1) = \prod_{i=1}^{m} c_i + 1$$

n 的所有正约数的和为：

$$(1 + p_1 + p_2^2 + \cdots + p_1^{c_1}) * \cdots * (1 + p_m + p_m^2 + \cdots + p_m^{c_m}) = \prod_{i=1}^{m} (\sum_{j=0}^{c_i} (p_i)^j)$$

求 n 的正约数

若 $d \geq \sqrt{n}$ 是 n 的约数，则 $n/d \leq \sqrt{n}$ 也是 n 的约数。换言之，约数总是成对出现的（除了对于完全平方数，$\sqrt{n}$ 会单独出现）。
因此，我们只需要扫描 $d = 1 \sim \sqrt{n}$，尝试 d 能够整除 n，若能整除，则 n/d 也是 n 的约数。时间复杂度为 $O(\sqrt{n})$。
通过这个"试除法"，我们可以得知，一个整数 n 的约数上界为 $2\sqrt{n}$。

求 n 的正约数

若 $d \geq \sqrt{n}$ 是 n 的约数，则 $n/d \leq \sqrt{n}$ 也是 n 的约数。换言之，约数总是成对出现的（除了对于完全平方数，$\sqrt{n}$ 会单独出现）。
因此，我们只需要扫描 $d = 1 \sim \sqrt{n}$，尝试 d 能够整除 n，若能整除，则 n/d 也是 n 的约数。时间复杂度为 $O(\sqrt{n})$。
通过这个“试除法”，我们可以得知，一个整数 n 的约数上界为 $2\sqrt{n}$。

如果我们使用上述“试除法”分别求 $1 \sim n$ 每个数的正约数集合，时间复杂度过高，为 $O(n\sqrt{n})$。可以反过来考虑，对于每个数 d，找出 $1 \sim n$ 中哪些数是 d 的倍数，$d, 2d, 3d, \cdots, \lfloor n/d \rfloor * d$ 这些数都有一个约数 d。
时间复杂度为 $O(n + n/2 + n/3 + \cdots + n/n) = O(n \log n)$。
通过这个“倍数法”，我们可以得知，$1 \sim n$ 每个数的约数个数的总和大约为 $n \log n$。

例题

反素数

对于任何正整数 x，其约数的个数记为 $g(x)$。例如：$g(1) = 1, g(6) = 4$。如果某个正整数 x 满足：对于任意的 $0 < i < x$，都有 $g(x) > g(i)$，那么称 x 为反质数。例如，整数 $1, 2, 4, 6$ 等都是反质数。
现在给定一个数 $n(1 \leq n \leq 2 * 10^9)$，请求出不超过 n 的最大的反质数。

例题

反素数

对于任何正整数 x，其约数的个数记为 $g(x)$。例如：$g(1)=1, g(6)=4$。
如果某个正整数 x 满足：对于任意的 $0<i<x$，都有 $g(x)>g(i)$，那么称 x 为反质数。例如，整数 $1,2,4,6$ 等都是反质数。
现在给定一个数 $n(1\leq n\leq 2*10^9)$，请求出不超过 n 的最大的反质数。

引理 1：$1\sim n$ 中最大的反质数，就是 $1\sim n$ 中约数个数最多的数中最小的一个。
证明：设 m 是 $1\sim n$ 中约数个数最多的数中最小的一个，则有：

- $\forall x<m, g(x)<g(m)$
- $\forall x>m, g(x)\geq g(m)$

第一条说明 m 是反质数，第二条说明大于 m 的数都不是反质数，故 m 即为所求。

反素数

引理 2：$1 \sim n$ 中任何数的不同质因子都不会超过 10 个，且所有质因子指数总和不超过 30。
证明：最小的 11 个质数的乘积为 $2*3*5*7*11*13*17*19*23*29*31 > 2*10^9$，所以 $n \le 2*10^9$ 不可能有多于 10 个不同的质因子。

引理 3：$\forall x \in [1,n]$，x 为反质数的必要条件是：x 分解质因数后可写作 $2^{c_1}*3^{c_2}*5^{c_3}*7^{c_4}*11^{c_5}*13^{c_6}*17^{c_7}*19^{c_8}*23^{c_9}*29^{c_{10}}$，并且 $c_1 \ge c_2 \cdots c_{10} \ge 0$。也就是说，$x$ 的质因数是连续的若干个最小的质数，并且指数非严格单调递减。

反素数

引理 2：$1 \sim n$ 中任何数的不同质因子都不会超过 10 个，且所有质因子指数总和不超过 30。
证明：最小的 11 个质数的乘积为 $2*3*5*7*11*13*17*19*23*29*31 > 2*10^9$，所以 $n \le 2*10^9$ 不可能有多于 10 个不同的质因子。

引理 3：$\forall x \in [1, n]$，x 为反质数的必要条件是：x 分解质因数后可写作 $2^{c_1} * 3^{c_2} * 5^{c_3} * 7^{c_4} * 11^{c_5} * 13^{c_6} * 17^{c_7} * 19^{c_8} * 23^{c_9} * 29^{c_{10}}$，并且 $c_1 \ge c_2 \cdots c_{10} \ge 0$。也就是说，$x$ 的质因数是连续的若干个最小的质数，并且指数非严格单调递减。

综上所述，我们可以通过 DFS 来尝试依次确定前 10 个质数的指数，并满足指数单调递减、总乘积不超过 n, 同时记录约数的个数。
在这两个限制条件先，搜索量实际上非常小。每次搜索出一个满足条件的整数，我们就按照引理 1 更新答案。

例题

余数之和

给定正整数 n 和 k，计算 $(k \bmod 1 + (k \bmod 2) + \cdots + (k \bmod n)$ 的值。
$1 \leq n, k \leq 10^9$。

例题

余数之和

给定正整数 n 和 k，计算 $(k \bmod 1 + (k \bmod 2) + \cdots + (k \bmod n)$ 的值。
$1 \leq n, k \leq 10^9$。

注意到 $k \bmod i = k - \lfloor k/i \rfloor * i$，故可以转化为计算 $n * k - \sum_{i=1}^{n} \lfloor k/i \rfloor * i$。
对于任意正整数 $x \in [1, k]$，设 $g(x) = \lfloor k / \lfloor k/x \rfloor \rfloor$。函数 $f(x) = k/x$ 单调递减，而 $g(x) \geq \lfloor k/(k/x) \rfloor = x$，故 $\lfloor k/g(x) \rfloor \leq \lfloor k/x \rfloor$。
另外，$\lfloor k/g(x) \rfloor \geq \lfloor k/(k/\lfloor k/x \rfloor) \rfloor = \lfloor k/k * \lfloor k/x \rfloor \rfloor = \lfloor k/x \rfloor$。
故 $\lfloor k/g(x) \rfloor = \lfloor k/x \rfloor$。进一步可得，$\forall i$ 满足 $x \leq i \leq \lfloor k / \lfloor k/x \rfloor \rfloor$，$\lfloor k/i \rfloor$ 的值都相等。

例题

余数之和

给定正整数 n 和 k，计算 $(k \bmod 1 + (k \bmod 2) + \cdots + (k \bmod n)$ 的值。
$1 \le n, k \le 10^9$。

注意到 $k \bmod i = k - \lfloor k/i \rfloor * i$，故可以转化为计算 $n * k - \sum_{i=1}^{n} \lfloor k/i \rfloor * i$。
对于任意正整数 $x \in [1, k]$，设 $g(x) = \lfloor k / \lfloor k/x \rfloor \rfloor$。函数 $f(x) = k/x$ 单调递减，而 $g(x) \ge \lfloor k/(k/x) \rfloor = x$，故 $\lfloor k/g(x) \rfloor \le \lfloor k/x \rfloor$。
另外，$\lfloor k/g(x) \rfloor \ge \lfloor k/(k/\lfloor k/x \rfloor) \rfloor = \lfloor k/k * \lfloor k/x \rfloor \rfloor = \lfloor k/x \rfloor$。
故 $\lfloor k/g(x) \rfloor = \lfloor k/x \rfloor$。进一步可得，$\forall i$ 满足 $x \le i \le \lfloor k / \lfloor k/x \rfloor \rfloor$，$\lfloor k/i \rfloor$ 的值都相等。
所以 $\forall i \in [1, k]$，$\lfloor k/i \rfloor$ 至多只有 $2\sqrt{k}$ 个不同的值。这是因为当 $i \le \sqrt{k}$ 时，i 只有 $\sqrt{n}$ 种选择，故 $\lfloor k/i \rfloor$ 至多只有 $\sqrt{k}$ 个不同的值；而当 $i > \sqrt{k}$ 时，$\lfloor k/i \rfloor < \sqrt{k}$，$\lfloor k/i \rfloor$ 也至多只有 $\sqrt{k}$ 个不同的值。

余数之和

综上所述，对于 $i = 1 \sim k$，$\lfloor k/i \rfloor$ 由不超过 $2\sqrt{k}$ 段组成，每一段 $x \le i \le \lfloor k/\lfloor k/x \rfloor \rfloor$ 种 $\lfloor k/i \rfloor$ 的值都等于 $\lfloor k/x \rfloor$。在该段中，$\lfloor k/i \rfloor * i = \lfloor k/x \rfloor * i$ 是一个等差数列，可以直接使用进行求和计算。
整个算法的复杂度为 $O(\sqrt{k})$。
核心代码如下：

```
ans=n*k;
for(int x=1,gx;x<=n;x=gx+1){
  gx=k/x ? min(k/(k/x),n) : n;
  ans-=(k/x)*(x+gx)*(gx-x+1)/2;
}
```

最大公约数

定义:
若自然数 d 同时是自然数 a 和 b 的约数，则称 d 是 a 和 b 的公约数。在所有 a 和 b 的公约数中最大的一个，称为 a 和 b 的最大公约数，记为 $gcd(a,b)$。
若自然数 m 同时是自然数 a 和 b 的倍数，则称 m 是 a 和 b 的公倍数。在所有 a 和 b 的公倍数中最小的一个，称为 a 和 b 的最小公倍数，记为 $lcm(a,b)$。
同理，我们也可以定义三个数以及更多个数的最大公约数、最小公倍数。

定理:

$$\forall a,b \in \mathbb{N}, \quad gcd(a,b) * lcm(a,b) = a * b$$

证明:
设 $d = gcd(a,b), a_0 = a/d, b_0 = b/d$，则有 $gcd(a_0,b_0) = 1$。根据最小公倍数的定义，有 $lcm(a_0,b_0) = a_0 * b_0$。于是有:

$$lcm(a,b) = lcm(a_0 * d, b_0 * d) = lcm(a_0,b_0) * d = a_0 * b_0 * d = a * b/d$$

更相减损术

$\forall a, b \in \mathbb{N}, a \geq b$，有 $gcd(a, b) = gcd(b, a - b) = gcd(a, a - b)$
$\forall a, b \in \mathbb{N}$，有 $gcd(2a, 2b) = 2gcd(a, b)$
证明:
根据定义，后者显然得证。我们尝试来证明前者。
对于 a, b 的任意公约数 d，因为 $d|a, d|b$，所以 $d|(a - b)$。因此 d 也是 $b, (a - b)$ 的公约数。反之亦成立。故 a, b 的公约数集合与 $b, (a - b)$ 的公约数集合相等，于是它们的最大公约数也相等。

欧几里得算法

$$\forall a, b \in \mathbb{N}, b \neq 0, \quad gcd(a, b) = gcd(b, a \bmod b)$$

证明:
若 $a < b$，则 $gcd(b, a \bmod b) = gcd(b, a) = gcd(a, b)$，命题成立。
若 $a \geq b$，设 $a = q * b + r$，其中 $0 \leq r < b$。显然 $r = a \bmod b$。对于 a, b 的任意公约数 d，因为 $d|a, d|q * b$，所以 $d|(a - qb)$，即 $d|r$，因此 d 也是 b, r 的公约数。反之亦成立。故 a, b 的公约数集合与 $b, a \bmod b$ 的公约数集合相同，它们的最大公约数自然也相等。
核心代码如下:

```
int gcd(int a,int b){
  return b ? gcd(b,a%b) : a;
}
```

该算法的时间复杂度为 $O(\log(a + b))$。

例题

Hankson 的趣味题

有 n 个询问，每次给定四个自然数 a, b, c, d，然后求有多少个 x 满足 $gcd(a, x) = c$ 且 $lcm(b, x) = d$。$n \leq 2000, 1 \leq a, b, c, d \leq 2 * 10^9$。

例题

Hankson 的趣味题

有 n 个询问，每次给定四个自然数 a, b, c, d，然后求有多少个 x 满足 $gcd(a, x) = c$ 且 $lcm(b, x) = d$。$n \leq 2000, 1 \leq a, b, c, d \leq 2 * 10^9$。

通过 $lcm(b, x) = d$ 可以得知 x 是 d 的约数。所以一个比较显然的想法是，枚举 d 的所有约数，依次判断是否满足两个条件。时间复杂度为 $O(n\sqrt{d}\log d)$，无法通过本题。
虽然 d 的约数个数上界大约是 $\sqrt{n}$，但是 $1 \sim d$ 中平均每个数的约数只有 $\log d$ 个。实际上，$2 * 10^9$ 之内的自然数中，约数个数最多的自然数仅有 1536 个约数。
所以们可以在进行试除法的时候使用之前的优化，预处理出所有质数后只枚举所有质数，先进行质因数分解，再找出所有约数，之后进行判断，这样就能获得满分。

Hankson 的趣味题

当然，我们也有复杂度更正确的做法。
因为 x 是 d 的约数，所以 x 的质因子一定也是 d 的质因子。我们可以对 d 的每个质因子 p，计算 x 可能包含多少个 p。
设 a, b, c, d, x 分别包含 m_a, m_b, m_c, m_d, m_x 个质因子 p，其中 m_x 是未知量。
由于 $gcd(a, x) = c$，故 $m_a \geq m_c$，且当 $m_a > m_c$ 时，$m_x = m_c$。
由于 $lcm(b, x) = d$，故 $m_b \leq m_d$，且当 $m_b < m_d$ 时，$m_x = m_d$。
只有当 $m_a = m_c, m_b = m_d$ 时，$m_c \leq m_x \leq m_d$。当 $m_a > m_c, m_b < m_d$ 且 $m_c \neq m_d$ 时，m_x 无解。
我们把 m_x 可能的取值数量记为 cnt_p，也就是 x 包含质因数 p 的方案有 cnt_p 种。根据乘法原理，满足题意的 x 数量即为：

$$\prod_{p|d, p\in Prime} cnt_p$$

先预处理出 $1 \sim \sqrt{2 * 10^9}$ 之间的质数，之后只枚举质数来进行质因数分解。时间复杂度为 $O(n\sqrt{d}/\log d)$。

互质与欧拉函数

$\forall a, b \in \mathbb{N}$，若 $gcd(a.b) = 1$，则称 a, b 互质。
对于三个数或者更多数的情况，我们把 $gcd(a, b, c) = 1$ 的情况称为 a, b, c 互质。把 $gcd(a, b) = gcd(b, c) = gcd(a, c) = 1$ 称为 a, b, c 两两互质。显然，如果三个数两两互质，它们肯定互质。

互质与欧拉函数

$\forall a, b \in \mathbb{N}$，若 $gcd(a.b) = 1$，则称 a, b 互质。
对于三个数或者更多数的情况，我们把 $gcd(a, b, c) = 1$ 的情况称为 a, b, c 互质。把 $gcd(a, b) = gcd(b, c) = gcd(a, c) = 1$ 称为 a, b, c 两两互质。显然，如果三个数两两互质，它们肯定互质。

$1 \sim n$ 中与 n 互质的数的个数被称为欧拉函数，记为 $\varphi(n)$。
若在算数基本定理中，$n = p_1^{c_1} p_2^{c_2} \cdots p_m^{c_m}$，则：

$$\varphi(n) = n * \frac{p_1 - 1}{p_1} * \frac{p_2 - 1}{p_2} * \cdots * \frac{p_m - 1}{p_m} = n * \prod_{p|n, p \in Prime} (1 - \frac{1}{p})$$

证明：设 p 是 n 的质因子，$1 \sim n$ 中 p 的倍数有 n/p 个。同理，若 q 也是 n 的质因子，$1 \sim n$ 中 p 的倍数有 n/q 个。如果我们把这些数减掉，那么是 pq 倍数的数被减掉了两次，需要加回来 $n/(pq)$ 个。也就是：

$$n - \frac{n}{p} - \frac{n}{q} + \frac{n}{pq} = n(1 - \frac{1}{p} - \frac{1}{q} + \frac{1}{pq}) = n(1 - \frac{1}{p})(1 - \frac{1}{q})$$

根据这个公式，可以在对 n 进行质因数分解时，顺便计算欧拉函数。

欧拉函数的性质

- $\forall n > 1$，$1 \sim n$ 中与 n 互质的数的和为 $n * \varphi(n)/2$。
- 若 a, b 互质，则 $\varphi(ab) = \varphi(a)\varphi(b)$。

证明：因为 $gcd(n, x) = gcd(n, n - x)$，所以与 n 不互质的数 $x, n - x$ 成对出现，平均值为 $n/2$。因此与 n 互质的数的平均值也是 $n/2$，进而得到第一条性质。
而根据欧拉函数的计算式，对 a, b 分解质因数，可以直接得到第二条性质。实际上，我们可以把第二条性质推广到更一般的函数上，可以得到积性函数的概念：
如果当 a, b 互质时，有 $f(ab) = f(a) * f(b)$，那么称函数 f 为积性函数。

欧拉函数的性质

- 若 f 是积性函数，且 $n = \prod_{i=1}^{m} p_i^{c_i}$，则 $f(n) = \prod_{i=1}^{m} f(p_i^{c_i})$。
- 设 p 是质数，若 $p|n$ 且 $p^2|n$，则 $\varphi(n) = \varphi(n/p) * p$。
- 设 p 是质数，若 $p|n$ 且 $p^2 \nmid n$，则 $\varphi(n) = \varphi(n/p) * (p-1)$。
- $\sum_{d|n} \varphi(d) = n$

按照积性函数的定义，第一条显然成立。
把 n 进行质因数分解并计算 φ 值，二者相除，发现商为 p，成立。
对于第三条，由于 $p, n/p$ 互质，由 φ 是积性函数可以推导出来。
对于第四条，设 $f(n) = \sum_{d_n} \varphi(d)$，若 n, m 互质，则：

$$f(nm) = \sum_{d|nm} \varphi(d) = (\sum_{d|n} \varphi(d))(\sum_{d|m} \varphi(d)) = f(n) * f(m)$$

所以 f 是积性函数。而对于单个质因子：
$f(p^m) = \sum_{d|p^m} \varphi(d) = \varphi(1) + \varphi(p) + \varphi(p^2) + \cdots + \varphi(p^m) = p^m$
所以 $f(n) = \prod_{i=1}^{m} f(p_i^{c_i}) = \prod_{i=1}^{m} p_i^{c_i} = n$。

例题

Visible Lattice Points

在一个平面直角坐标系的以 $(0,0)$ 为左下角，(n,n) 为右上角的矩形中，除了 $(0,0)$ 外，每个横、纵坐标都为整数的位置上都插着一个钉子。求在原点向四周看去，能够看到多少个钉子。一个钉子能被看到，当且仅当连接它和原点的线段上没有其它钉子。

$1 \leq n \leq 1000$。

例题

Visible Lattice Points

在一个平面直角坐标系的以 $(0,0)$ 为左下角，(n,n) 为右上角的矩形中，除了 $(0,0)$ 外，每个横、纵坐标都为整数的位置上都插着一个钉子。
求在原点向四周看去，能够看到多少个钉子。一个钉子能被看到，当且仅当连接它和原点的线段上没有其它钉子。
$1 \leq n \leq 1000$。

分析题目可以发现，除了 $(1,0),(0,1),(1,1)$ 三个钉子以外，一个钉子 (x,y) 能被看到，当且仅当 $1 \leq x, y \leq n, x \neq y$ 且 $gcd(x,y) = 1$。
而在 $1 \leq x, y \leq n, x \neq y$ 中能看到的钉子关于直线 $y = x$ 对称，所以我们可以只考虑其中一半，也就是 $1 \leq x < y \leq n$。换言之，我们需要对于每个 $2 \leq y \leq n$，统计有多少个 x 满足 $1 \leq x < y$ 且 $gcd(x,y) = 1$，发现这就是 $\varphi(y)$。
综上所述，本题的答案就是 $3 + 2 * \sum_{i=1}^{n} \varphi(i)$。我们要做的，就是求出 $2 \sim n$ 中每个数的欧拉函数。

Visible Lattice Points

我们可以使用埃氏筛，按照欧拉函数的计算式在 $O(n\log n)$ 的时间内求出每个数的欧拉函数：

```
void euler(int n){
  for(int i=2;i<=n;i++) phi[i]=i;
  for(int i=2;i<=n;i++)
    if(phi[i]==i)
      for(int j=i;j<=n;j+=i)
        phi[j]=phi[j]/i*(i-1);
}
```

不过，我们还可以利用之前的两条性质

- 设 p 是质数，若 $p|n$ 且 $p^2|n$，则 $\varphi(n)=\varphi(n/p)*p$。
- 设 p 是质数，若 $p|n$ 且 $p^2\nmid n$，则 $\varphi(n)=\varphi(n/p)*(p-1)$。

来在线性时间复杂度内筛出欧拉函数。

Visible Lattice Points

```
void euler(int n){
  memset(v,0,sizeof(v));  //最小质因子
  m=0;     //质数数量
  for(int i=2;i<=n;i++){
    if(v[i]==0){  //i 是质数
      v[i]=i;prime[++m]=i;
      phi[i]=i-1;
    }
    //给当前的数 i 乘上一个质因子
    for(int j=1;j<=m;j++){
      //i 有比 prime[j] 更小的质因子，或超出 n 的范围，停止循环
      if(prime[j]>v[i] || prime[j]>n/i) break;
      //prime[j] 是合数 i*prime[j] 的最小质因子
      v[i*prime[j]]=prime[j];
      phi[i*prime[j]]=
        phi[i]*(i%prime[j] ? prime[j]-1 : prime[j]);
      //判断 i,prime[j] 是否互质
    }
  }
}
```