

欧拉函数

定义

欧拉函数 (Euler's totient function)，即 $\varphi(n)$ ，表示的是小于等于 n 和 n 互质的数的个数。

比如说 $\varphi(1) = 1$ 。

当 n 是质数的时候，显然有 $\varphi(n) = n - 1$ 。

性质

- 欧拉函数是 [积性函数](#)。

即对任意满足 $\gcd(a, b) = 1$ 的整数 a, b ，有 $\varphi(ab) = \varphi(a)\varphi(b)$ 。

特别地，当 n 是奇数时 $\varphi(2n) = \varphi(n)$ 。

证明参见 [剩余系的复合](#)。

- $n = \sum_{d|n} \varphi(d)$ 。

证明

利用 [莫比乌斯反演](#) 相关知识可以得出。

也可以这样考虑：如果 $\gcd(k, n) = d$ ，那么 $\gcd(\frac{k}{d}, \frac{n}{d}) = 1, (k < n)$ 。

如果我们设 $f(x)$ 表示 $\gcd(k, n) = x$ 的数的个数，那么 $n = \sum_{i=1}^n f(i)$ 。

根据上面的证明，我们发现， $f(x) = \varphi(\frac{n}{x})$ ，从而 $n = \sum_{d|n} \varphi(\frac{n}{d})$ 。注意到约数 d 和 $\frac{n}{d}$ 具有对称性，所以上式化为 $n = \sum_{d|n} \varphi(d)$ 。

- 若 $n = p^k$ ，其中 p 是质数，那么 $\varphi(n) = p^k - p^{k-1}$ 。（根据定义可知）
- 由唯一分解定理，设 $n = \prod_{i=1}^s p_i^{k_i}$ ，其中 p_i 是质数，有 $\varphi(n) = n \times \prod_{i=1}^s \frac{p_i - 1}{p_i}$ 。

证明

- 引理：设 p 为任意质数，那么 $\varphi(p^k) = p^{k-1} \times (p - 1)$ 。

证明：显然对于从 1 到 p^k 的所有数中，除了 p^{k-1} 个 p 的倍数以外其它数都与 p^k 互素，故 $\varphi(p^k) = p^k - p^{k-1} = p^{k-1} \times (p - 1)$ ，证毕。

接下来我们证明 $\varphi(n) = n \times \prod_{i=1}^s \frac{p_i - 1}{p_i}$ 。由唯一分解定理与 $\varphi(x)$ 函数的积性

$$\begin{aligned}\varphi(n) &= \prod_{i=1}^s \varphi(p_i^{k_i}) \\ &= \prod_{i=1}^s (p_i - 1) \times p_i^{k_i-1} \\ &= \prod_{i=1}^s p_i^{k_i} \times \left(1 - \frac{1}{p_i}\right) \\ &= n \prod_{i=1}^s \left(1 - \frac{1}{p_i}\right) \quad \square\end{aligned}$$

- 对任意不全为 0 的整数 m, n ， $\varphi(mn)\varphi(\gcd(m, n)) = \varphi(m)\varphi(n)\gcd(m, n)$ 。

可由上一条直接计算得出。

实现

如果只要求一个数的欧拉函数值，那么直接根据定义质因数分解的同时求就好了。这个过程可以用 [Pollard Rho](#) 算法优化。

参考实现

C++

```
1  #include <cmath>
2
3  int euler_phi(int n) {
4      int ans = n;
5      for (int i = 2; i * i <= n; i++)
6          if (n % i == 0) {
7              ans = ans / i * (i - 1);
8              while (n % i == 0) n /= i;
9          }
10     if (n > 1) ans = ans / n * (n - 1);
11     return ans;
12 }
```

Python

```
1  import math
2
3
4  def euler_phi(n):
5      ans = n
6      for i in range(2, math.isqrt(n) + 1):
7          if n % i == 0:
8              ans = ans // i * (i - 1)
9              while n % i == 0:
10                 n = n // i
11     if n > 1:
12         ans = ans // n * (n - 1)
13     return ans
```

如果是多个数的欧拉函数值，可以利用后面会提到的线性筛法来求得。

详见：[筛法求欧拉函数](#)

应用

欧拉函数常常用于化简一系列最大公约数的和。国内有些文章称它为 **欧拉反演**¹。

在结论

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

中代入 $n = \gcd(a, b)$ ，则有

$$\gcd(a, b) = \sum_{d|\gcd(a, b)} \varphi(d) = \sum_d [d|a][d|b]\varphi(d),$$

其中， $[\cdot]$ 称为 **Iverson 括号**，只有当命题 P 为真时 $[P]$ 取值为 1，否则取 0。对上式求和，就可以得到

$$\sum_{i=1}^n \gcd(i, n) = \sum_d \sum_{i=1}^n [d|i][d|n]\varphi(d) = \sum_d \left\lfloor \frac{n}{d} \right\rfloor [d|n]\varphi(d) = \sum_{d|n} \left\lfloor \frac{n}{d} \right\rfloor \varphi(d).$$

这里关键的观察是 $\sum_{i=1}^n [d|i] = \left\lfloor \frac{n}{d} \right\rfloor$ ，即在 1 和 n 之间能够被 d 整除的 i 的个数是 $\left\lfloor \frac{n}{d} \right\rfloor$ 。

利用这个式子，就可以遍历约数求和了。需要多组查询的时候，可以预处理欧拉函数的前缀和，利用数论分块查询。

GCD SUM

给定 $n \leq 100000$ ，求

$$\sum_{i=1}^n \sum_{j=1}^n \gcd(i, j).$$

思路

仿照上文的推导，可以得出

$$\sum_{i=1}^n \sum_{j=1}^n \gcd(i, j) = \sum_{d=1}^n \left\lfloor \frac{n}{d} \right\rfloor^2 \varphi(d).$$

此时需要从 1 遍历到 n 求欧拉函数，用线性筛做就可以 $O(n)$ 得到答案。

欧拉定理

与欧拉函数紧密相关的一个定理就是欧拉定理。其描述如下：

若 $\gcd(a, m) = 1$ ，则 $a^{\varphi(m)} \equiv 1 \pmod{m}$ 。

扩展欧拉定理

当然也有扩展欧拉定理，用于处理一般的 a 和 m 的情形。

$$a^b \equiv \begin{cases} a^{b \bmod \varphi(m)}, & \gcd(a, m) = 1 \\ a^b, & \gcd(a, m) \neq 1, b < \varphi(m) \\ a^{b \bmod \varphi(m) + \varphi(m)}, & \gcd(a, m) \neq 1, b \geq \varphi(m) \end{cases} \pmod{m}$$

证明和习题详见 [欧拉定理](#)。


习题


- [SPOJ ETF. Euler Totient Function](#)
- [UVa 10179. Irreducible Basic Fractions](#)
- [UVa 10299. Relatives](#)
- [UVa 11327. Enumerating Rational Numbers](#)
- [TIMUS 1673. Admission to Exam](#)
- [Luogu P1390 公约数的和](#)
- [Luogu P2155 \[SDOI2008\] 沙拉公主的困惑](#)
- [Luogu P2568 GCD](#)

参考资料与注释

1. 这一说法并未见于学术期刊或国外的论坛中，在使用该说法时应当注意。 [←](#)

 本页面最近更新：2025/8/18 01:16:12，[更新历史](#)

 发现错误？想一起完善？ [在 GitHub 上编辑此页！](#)

 本页面贡献者：[lr1d](#), [guodong2005](#), [sshwy](#), [Tiphereth-A](#), [Xeonacid](#), [Enter-tainer](#), [iamtwz](#), [MegaOwler](#), [StudyingFather](#), [c-forrest](#), [Chrogeek](#), [mgt](#), [shuzhouliu](#), [aofall](#), [CCXXI](#), [CoelacanthusHex](#), [frank-xjh](#), [Great-designer](#), [greyqz](#), [henrytbtrue](#), [kZime](#), [lihaoyu1234](#), [Marcythm](#), [Menci](#), [nalemy](#), [orzAtalod](#), [ouuan](#), [Persdre](#), [segment-tree](#), [ShaoChenHeng](#), [Struggler-q](#), [yuhuoji](#), [ksyx](#), [Pinghigh](#), [shawlleyw](#), [TrisolarisHD](#), [TrisolarisHD](#)

© 本页面的全部内容 [在 CC BY-SA 4.0 和 SATA 协议之条款下](#) 提供，附加条款亦可能应用