素数判定与因数分解

摘要：给出一个不超过 2^{64} 的整数，判定其是否为质数以及将其因数分解.这类问题可以用 Miller-Rabin 测试和 Pollard-rho 因数分解算法处理.

目录

费马小定理:

如果 p 是素数, a 是小于 p 的正整数，那么 $a^{p-1} \equiv 1 \bmod p$**.**

费马小定理的逆命题:

如果不存在正整数 a 使得 $a^{p-1} \equiv 1 \bmod p$ **成立，那么 p 为素数.**

但是费马小定理的逆命题是错误的，一个常见的例子是 341，虽然他满足

$2^{340} \equiv 1 \bmod 341$，但是它不是素数.这类满足这个形式但是不是素数的数还有像 561, 645, 1105. 于是将满足 $a^{p-1} \equiv 1 \bmod p$ 的数 p 称为是以 a 为底的**伪素数**.

对于一个 a，不满足 $a^{p-1} \equiv 1 \bmod p$ 的数 p 一定是合数，如果满足的话有很大可能是质数. 统计发现，如果数 p 满足 $2^{p-1} \equiv 1 \bmod p$，那么 p 有大约 99.9%的几率是素数，如果 p 同时满足 $2^{p-1} \equiv 1 \bmod p$ 和 $3^{p-1} \equiv 1 \bmod p$，那么大约有 99.97%的概率是素数.也就是说选择的 a 越多，错误率越低.

是不是说明如果选取的 a 足够多，准确率就能达到 100%呢？已经证实，存在一种数，能通过所有 a 的测试. $2^{p-1} \equiv 1 \bmod p$，这种数被称为 Carmichael 数. 第一个 Carmichael 数是 561, Carmichael 数有无限多个.

Miller-Rabin 强伪素数测试

考虑素数的这样一个性质：如果 p 为素数，那么 1 对模 p 的平方根[模平方根]只可能是 1 或-1. 因此根据费马小定理 $a^{p-1} \equiv 1 \bmod p$，a^{p-1} 对模 p 的平方根 $a^{\frac{n-1}{2}}$ 也只能是 1 和-1. 如果 $\frac{n-1}{2}$ 本身是一个偶数，那么可以对其再取一次平方根......

将上述步骤写成一个算法就是 Miller-Rabin 强伪素数测试:

记 $p-1=2^b d$，其中 d 为奇数. 那么如果 $a^d \equiv 1 \bmod p$ 或者存在 $1 \le r \le d$ 使得 $a^{2^r d} \equiv 1 \bmod p$.就认为 p 通过测试.

如果通过测试的 p 为合数，就称之为是一个以 a 为基的**强伪素数**.一次测试的错误率不超过 $\frac{1}{4}$. 但是如果进行多次测试，错误率就会被降到比较低.

在具体实现时可以将这个测试简答的替换为:

如果 p 满足 $a^{p-1} \equiv 1 \bmod p$，那么通过测试.

Birthday Trick

Birthday Trick 是一个提高概率的技巧.举例说明.

从[1,1000]中任取一个数取到某个数的概率为$\frac{1}{1000}$.如果从[1,1000]中任取两个数其差值为某个数的概率为$\frac{1}{500}$.如果在[1,1000]中任取 k 个数其中有两个数的差值为某个数的概率为多少呢? 验证得到当 k=30 时概率超过 50%.

生日悖论

在一个超过 23 个人的屋子里，有两个人生日相同的概率超过 50%.

生日悖论可以看做是在[1,365]中任取 n 个数存在相同数的概率的问题.

n 个人生日不存在相同的概率为 $p=\sum_{i=0}^{n-1}\frac{365-i}{365}$. 经计算，23 个人存在生日相同的概率大约为 50.7%.如果一年中有 N 天，那么每 $k=\sqrt{N}$ 个人中有超过 50%的概率产生生日冲突.

现在就可以利用生日悖论来因数分解，要找到对于 $N=pq$ 的 N 的因子 p 和 q.

对于一个数N，在[1,N]中任意选取$\sqrt{N}$个数，记为$x_1,\cdots,x_k$，那么有超过50%的概率使得存在$x_i-x_j\mid N$. 那么对于这$\sqrt{N}$个数每两个数之间都要进行一次乘法. 其效率和从 1 到 N 一个一个试除没什么区别.

于是便可以这么做: 选取k个数，如果存在$(x_i-x_j,N)>1$，说明(x_i-x_j,N)是N的一个因子. 因为满足条件的x_i-x_j为$p,2p,\cdots,(q-1)p,q,2q,\cdots,(p-1)q$.因此大概需要选取$N^{\frac{1}{4}}$个数.对于较小的 N 的范围是可行的.

Pollard's Rho 算法

Pollard’s Rho 算法就是对于上述算法的改进，从而减少了空间消耗，成为一种可行的因数分解算法.它并不随机生成 k 个数并两两进行比较，而是一个一个地生成并检查连续的两个数.它利用了一个特殊的函数：$f(x)=x^2+a(\bmod N)$.其中 a 为一个常数.

我们从 $x_1=2$ 或者其他数开始 ,让 $x_n=f(x_{n-1})$. 计算 (x_i-x_{i-1},N) 是否大于 1，如果大于 1，说明 (x_i-x_{i-1},N) 是 N 的一个因子. 反之继续进行下去.

但是会有一种情况会使得存在某个 $x_j=x_i$，这样就会在若干次后回到起点，称其出现了环. 如何检测是否出现了环呢?

有一个简单的方法来判定，它是由 Floyd 发现的，就是如果存在一个环，那么让 a,b 从同一起点出发, b 以两倍于 a 的速度往前走，当 a 与 b 相遇时存在着环.

这样就得到了基于 Floyd 的周期检测策略的 Pollard’s Rho 算法了.

Reference

[1] A Quick Tutorial on Pollard's Rho Algorithm, Computer Science of Colorado University.

[2] int64 内 Miller-Rabin 素数测试和 Pollard_Rho_因数分解算法实现.