Testes de primalidade: probabilísticos e determinísticos

Carlos Gustavo T. de A. Moreira, Nicolau C. Saldanha

25 de abril de 2011

> O problema de distinguir números primos de compostos e de decompor esses últimos em seus fatores primos é conhecido como sendo um dos mais importantes e úteis na aritmética. . . . . a dignidade da própria ciência parece requerer que todos os meios possíveis sejam explorados para a solução de um problema tão elegante e tão celebrado.
> *Karl Friedrich Gauß, Disquisitiones Arithmeticae, 1801*

1 Introdução

Não se conhece nenhuma fórmula simples para gerar primos arbitrariamente grandes. Uma palavra imprecisa mas importante nesta frase é "simples". Existem fórmulas que geram números primos, mas que são tão complicadas que não ajudam muito nem a gerar números primos explicitamente nem a responder perguntas teóricas sobre a distribuição dos primos. Um exemplo de fórmula para p_n, o n-ésimo primo, é

$$p_n = \left\lfloor 1 - \frac{1}{\log 2} \log \left(-\frac{1}{2} + \sum_{d|P_{n-1}} \frac{\mu(d)}{2^d - 1} \right) \right\rfloor,$$

onde $P_{n-1} = p_1 p_2 \cdots p_{n-1}$; aqui μ é função de Möbius: $\mu(n) = 0$ se n for múltiplo de algum quadrado de primo e $\mu(n) = (-1)^k$ se n for o produto de k primos distintos. Deixamos a demonstração a cargo do leitor. Outra fórmula é

$$p_n = \lfloor 10^{2^n} c \rfloor - 10^{2^{n-1}} \lfloor 10^{2^{n-1}} c \rfloor,$$

onde

$$c = \sum_{n=1}^{\infty} \frac{p_n}{10^{2^n}} = 0.0203000500000007\ldots.$$

A inutilidade desta última fórmula vem do fato que para calcular c devemos encontrar todos os primos; a fórmula se tornaria mais interessante se existisse outra interpretação para o número real c, o que parece muito improvável. Mills ([6]) também provou que existem números reais $A > 1$ tal que $\lfloor A^{3^n} \rfloor$ é primo para todo $n \in \mathbb{N}$.

Um tipo de fórmula para primos, de certa forma mais intrigante, são polinômios de coeficientes inteiros em S variáveis com a seguinte propriedade quase mágica: a intersecção da imagem de $\mathbb{N}^S$ com $\mathbb{N}$ é exatamente o conjunto dos números primos. Note que se tomarmos um ponto de $\mathbb{N}^S$ "ao acaso", o valor do polinômio neste ponto quase certamente será negativo; assim, é difícil usar o polinômio para gerar primos. A título de curiosidade, vejamos um exemplo de polinômio com estas

propriedades; aqui $S = 26$, o valor do polinômio é P, as variáveis chamam-se $a, b, \ldots, z$ e $A, B, \ldots, N$ são expressões auxiliares:

$$
\begin{aligned}
P &= (k+2)(1 - A^2 - B^2 - C^2 - \cdots - N^2),\\
A &= wz + h + j - q,\\
B &= (gk + 2g + k + 1)(h + j) + h - z,\\
C &= 16(k+1)^3(k+2)(n+1)^2 + 1 - f^2,\\
D &= 2n + p + q + z - e,\\
E &= e^3(e+2)(a+1)^2 + 1 - o^2,\\
F &= (a^2 - 1)y^2 + 1 - x^2,\\
G &= 16r^2y^4(a^2 - 1) + 1 - u^2,\\
H &= ((a + u^2(u^2 - a))^2 - 1)(n + 4dy)^2 + 1 - (x + cu)^2,\\
I &= (a^2 - 1)l^2 + 1 - m^2,\\
J &= ai + k + 1 - l - i,\\
K &= n + l + v - y,\\
L &= p + l(a - n - 1) + b(2an + 2a - n^2 - 2n - 2) - m,\\
M &= q + y(a - p - 1) + s(2ap + 2a - p^2 - 2p - 2) - x,\\
N &= z + pl(a - p) + t(2ap - p^2 - 1) - pm.
\end{aligned}
$$

Algumas observações simples: a única forma de P ser positivo é se $A = B = \cdots = N = 0$; neste caso seu valor será $k + 2$. Vemos assim que para produzir um número primo P com este polinômio devemos antes de mais nada tomar $k = P - 2$. As expressões auxiliares viram equações: como $A = 0$ temos $q = wz + h + j$. Assim, dado k para o qual $k + 2$ é primo, precisamos procurar valores para as outras letras que satisfaçam estas equações. Estes valores de certa forma *codificam* uma demonstração de que $P = k + 2$ é primo.

Uma questão relacionada com a de *gerar* números primos é a de *testar* se um determinado número é primo. Com o advento dos computadores, a partir da década de 60, surgiram inúmeras tentativas de se obter um algoritmo eficiente para o teste de primalidade de um número. A relevância desse problema tem crescido imensamente em anos recentes devido à utilização intensa de números primos em algoritmos de criptografia, como os algoritmos RSA e El Gammal para criptografia pública. Dessa forma o problema do teste de primalidade se tornou um importante problema para a ciência da computação teórica. Sobre esse ponto de vista duas coisas são requeridas: um certificado de prova de que o algoritmo realmente produz a resposta correta; e uma medida da eficiência do algoritmo, isto é, quão bem o algoritmo faz uso dos recursos computacionais (como o tempo ou número de passos executados, espaço ou memória utilizada) em função do tamanho da entrada do problema para a obtenção da solução.

Existe um algoritmo bastante simples, devido ao matemático grego Eratóstenes (ca. 240 A.C.), para testar se qualquer inteiro positivo n é primo: calcule o resto da divisão de n por cada inteiro m com $2 \leq m \leq \sqrt{n}$. Se o resto for 0 em algum caso então n é composto e encontramos um divisor; se isto nunca ocorrer, n é primo. O inconveniente deste algoritmo é que ele é muito lento. O tamanho da entrada do algoritmo para um dado número n é o tamanho da sua codificação em bits, que é aproximadamente $k = \log_2 n$ pois $2^k \leq n < 2^{k+1}$. Portanto, em termos do tamanho da entrada k, temos que o número de operações é $O(\sqrt{n}) = O(2^{k/2})$, ou seja, o algoritmo tem complexidade de tempo exponencial no tamanho da entrada. Veremos neste trabalho outros testes de primalidade mais rápidos que usam um pouco mais de teoria dos números.

2 Fundamentos

Dados dois inteiros d e a, dizemos que d *divide* a ou que d é um *divisor* de a ou ainda que a é um *múltiplo* de d e escrevemos

$$d \mid a$$

se existir $q \in \mathbb{Z}$ com $a = qd$. Sejam $a, n \in \mathbb{Z}$, $n > 0$: denotamos o resto da divisão de a por n por $a \bmod n$; em outras palavras, $a \bmod n$ é o único inteiro entre 0 e $n-1$ com $n \mid (a - (a \bmod n))$. Sejam $a, b, n \in \mathbb{Z}$. Dizemos que a *é congruente a* b *módulo* n, e escrevemos

$$a \equiv b \pmod{n}$$

se $n \mid a - b$, ou seja, se $(a \bmod n) = (b \bmod n)$. Por exemplo, temos que $17 \equiv 3 \pmod 7$ e $10 \equiv -5 \pmod 3$. Pelo algoritmo de Euclides sabemos que dados $a, b \in \mathbb{Z}$ existem $c, d \in \mathbb{Z}$ com $\text{mdc}(a, b) = ad - bc$. Se $\text{mdc}(a, n) = 1$ existe portanto $d \in \mathbb{Z}$ com $ad \equiv 1 \pmod n$ e dizemos que a é *invertível* módulo n.

Definimos a *função de Euler* φ por

$$\varphi(n) = |\{a \in \mathbb{Z}; 0 < a \leq n, \text{mdc}(a, n) = 1\}|$$

onde $n \in \mathbb{Z}$, $n > 0$ e $|X|$ denota o número de elementos de X. Temos $\varphi(1) = \varphi(2) = 1$, e, para $n > 2$, $1 < \varphi(n) < n$. Se p é primo, $\varphi(p) = p - 1$; mais geralmente $\varphi(p^k) = p^k - p^{k-1}$ pois $(a, p^k) = 1$ se e somente se a não é múltiplo de p e há p^{k-1} múltiplos de p no intervalo $0 \leq a < p^k$. Mais geralmente, temos

$$\varphi(n) = n \prod_{p|n} \left(1 - \frac{1}{p}\right).$$

Os $\varphi(n)$ números inteiros $b_1, b_2, \ldots, b_{\varphi(n)}$ formam um *sistema completo de invertíveis* (s.c.i.) módulo n se para todo $c \in \mathbb{Z}$, $\text{mdc}(c, n) = 1$, existe i, $0 < i < \varphi$, com $c \equiv b_i \pmod n$. Equivalentemente (pelo princípio da casa dos pombos), $b_1, b_2, \ldots, b_{\varphi(n)}$ formam um s.c.i. módulo n se e somente se $\text{mdc}(b_i, n) = 1$ para todo i e $b_i \equiv b_j \pmod n$ implicar $i = j$.

Lema 1 *Sejam $q, n \in \mathbb{Z}$, $n > 0$, $\text{mdc}(q, n) = 1$, e $b_1, b_2, \ldots, b_{\varphi(n)}$ um s.c.i. módulo n. Então os inteiros $qb_1, qb_2, ...qb_{\varphi(n)}$ também formam um s.c.i. módulo n.*

Demonstração: Como $\text{mdc}(q, n) = \text{mdc}(b_i, n) = 1$, temos $\text{mdc}(qb_i, n) = 1$. Por outro lado, se $qb_i \equiv qb_j \pmod n$ temos $b_i \equiv b_j \pmod n$ e $i = j$. □

Teorema 2 (Euler) *Sejam $a, n \in \mathbb{Z}$, $n > 0$, tais que $(a, n) = 1$. Então*

$$a^{\varphi(n)} \equiv 1 \pmod{n}.$$

Demonstração: Seja $b_1, b_2, \ldots, b_{\varphi(n)}$ um s.c.i. módulo n. Pela proposição anterior, os números $ab_1, ab_2, \ldots ab_{\varphi(n)}$ também formam um s.c.i. módulo n. Assim,

$$b_1 \cdot b_2 \cdots b_{\varphi(n)} \equiv ab_1 \cdot ab_2 \cdots ab_{\varphi(n)} \pmod{n}$$

pois módulo n os dois lados têm os mesmos fatores a menos de permutação. Mas isto pode ser reescrito como

$$a^{\varphi(n)}(b_1 \cdot b_2 \cdots b_{\varphi(n)}) \equiv 1 \cdot (b_1 \cdot b_2 \cdots b_{\varphi(n)}) \pmod{n}$$

e isto implica $a^{\varphi(n)} \equiv 1 \pmod n$. □

Corolário 3 (Fermat) *Se p é primo então, para todo inteiro a, $a^p \equiv a \pmod p$.*

DEMONSTRAÇÃO: Se $p \mid a$, então $a^p \equiv a \equiv 0 \pmod p$. Caso contrário, $\varphi(p) = p-1$, $a^{p-1} \equiv 1 \pmod p$ e novamente $a^p \equiv a \pmod p$. □

Se $a, n \in \mathbb{Z}$ com $n > 0$ e $\mathrm{mdc}(a,n) = 1$, definimos a *ordem de a módulo n*, denotado por $\mathrm{ord}_n a$, como o menor inteiro $t > 0$ tal que $a^t \equiv 1 \pmod n$.

Proposição 4 *Temos que $a^t \equiv 1 \pmod n$ se, e só se, $\mathrm{ord}_n a \mid t$. Em particular, para todo a com $\mathrm{mdc}(a,n) = 1$ temos $\mathrm{ord}_n a \mid \varphi(n)$.*

DEMONSTRAÇÃO: Como $a^{\mathrm{ord}_n a} \equiv 1 \pmod n$, para todo $k \in \mathbb{N}$ tem-se $a^{k\,\mathrm{ord}_n a} \equiv 1 \pmod n$. Por outro lado, se $a^t \equiv 1 \pmod n$, pelo algoritmo da divisão existem inteiros q e r tais que $0 \le r < \mathrm{ord}_n a$ e $t = q\,\mathrm{ord}_n a + r$. Portanto

$$1 \equiv a^t = a^{q\,\mathrm{ord}_n a + r} = (a^{\mathrm{ord}_n a})^q \cdot a^r \equiv a^r \pmod n.$$

Ou seja, $a^r \equiv 1 \pmod n$. Pela minimalidade de $\mathrm{ord}_n a$, temos que $r = 0$, i.e., $\mathrm{ord}_n a \mid t$. □

A proposição acima nem sempre dá a melhor estimativa para $\mathrm{ord}_n a$. Assim, por exemplo, para $n = 8$ temos $\varphi(n) = 4$ mas $\mathrm{ord}_n a \mid 2$ para todo a ímpar. Por outro lado, para n primo sempre existe uma *raiz primitiva* módulo n, isto é, um inteiro a com $\mathrm{ord}_n a = n - 1 = \varphi(n)$ ([3]). Por exemplo, 2 é raiz primitiva módulo 5, pois $\mathrm{ord}_5 2 = 4 = \varphi(5)$. Note que se a é raiz primitiva módulo p então $a^1, a^2, \ldots, a^{p-1} \equiv a^0 \equiv 1$ formam um s.c.i. módulo p.

Será importante em alguns casos discutir se a é um quadrado módulo n, i.e., se existe $b \in \mathbb{Z}$ com $b^2 \equiv a \pmod n$.

Proposição 5 *Seja $p > 2$, p primo.*

(a) Exatamente $\frac{p-1}{2}$ dentre os números $1, 2, \ldots, p-1$ são quadrados módulo p.

(b) Considere a com $\mathrm{mdc}(a,p) = 1$. Temos $a^{\frac{p-1}{2}} \equiv \pm 1 \pmod p$. Além disso, $a^{\frac{p-1}{2}} \equiv 1 \pmod p$ se e somente se a é quadrado módulo p.

DEMONSTRAÇÃO: Os quadrados (não nulos) módulo p são

$$1^2 \equiv (p-1)^2, 2^2 \equiv (p-2)^2, \ldots, (\frac{p-1}{2})^2 \equiv (\frac{p+1}{2})^2.$$

Note que cada quadrado aparece exatamente uma vez na lista acima pois $x^2 \equiv y^2 \pmod p$ implica $p \mid (x+y)(x-y)$ e portanto $x \equiv \pm y \pmod p$. Isto completa a prova do item (a).

Segue de Fermat que $(a^{\frac{p-1}{2}})^2 \equiv 1 \pmod p$. Ora, $x^2 \equiv 1 \pmod p$ implica $x \equiv \pm 1 \pmod p$. Assim $a^{\frac{p-1}{2}} \equiv \pm 1 \pmod p$. Se a é quadrado módulo p temos $a \equiv b^2$ e $a^{\frac{p-1}{2}} \equiv b^{p-1} \equiv 1 \pmod p$.

Resta demonstrar que se a não é quadrado módulo p então $a^{\frac{p-1}{2}} \equiv -1 \pmod p$. Uma forma de completar a demonstração é usar o fato que a congruência $X^{\frac{p-1}{2}} \equiv 1$ tem no máximo $\frac{p-1}{2}$ raízes módulo p (a demonstração é análoga a de que um polinômio de grau k tem no máximo k raízes reais; veja, por exemplo, [3]). Outra forma é usar a existência de uma raiz primitiva: se c é uma raiz primitiva módulo p temos $c^{\frac{p-1}{2}} \equiv -1 \pmod p$ (de fato, $c^{\frac{p-1}{2}} \equiv 1 \pmod p$ implica $\mathrm{ord}_p c \le \frac{p-1}{2}$ e c não é raiz primitiva). Mais geralmente, os não quadrados são da forma c^k, k ímpar. □

3 O teste probabilístico de Miller-Rabin

A ideia é usar o pequeno teorema de Fermat para testar a primalidade de n: tomamos a, $1 < a < n$, e calculamos $a^{n-1} \bmod n$. Se n for primo teremos $a^{n-1} \bmod n = 1$; qualquer outro resultado indica que n é composto mesmo sem termos encontrado um fator de n.

Observe que para calcular $a^m \bmod n$ não precisamos calcular $a \cdot a \cdots a$, m vezes. Se $m = \sum_{0 \le i < N} b_i 2^i$, $N = \lfloor \log_2 m \rfloor$, então definimos

$$e_k = \sum_{0 \le i < k} b_{N-k+i} 2^i, \quad c_k = a^{e_k} \bmod n$$

que pode ser reescrita recursivamente como

$$e_0 = 0, \quad e_{k+1} = 2e_k + b_{N-k},$$
$$c_0 = 1, \quad c_{k+1} = \begin{cases} c_k^2 \bmod n, & b_{N-k} = 0, \\ a c_k^2 \bmod n, & b_{N-k} = 1. \end{cases}$$

Fazendo as contas desta forma encontramos $c_N = a^m \bmod n$ com menos de $4 \log_2 m$ operações envolvendo inteiros menores do que n^2.

Se $a^{n-1} \bmod n = 1$, por outro lado, não demonstramos que n é primo; se n for composto satisfazendo $a^{n-1} \equiv 1 \pmod n$ dizemos que n é um *pseudoprimo* na base a. Pseudoprimos existem mas são raros (ver [4]): o menor pseudoprimo na base 2 é $341 = 11 \cdot 31$ e existem apenas 21 853 pseudoprimos na base 2 menores do que $2,5 \cdot 10^{10}$ (contra 1 091 987 405 primos). Pomerance (melhorando um resultado anterior de Erdős) provou que se $P\pi_a(x)$ é o número de pseudoprimos até x na base a temos

$$P\pi_a(x) \le x \cdot e^{-\frac{\log x \log\log\log x}{2 \log\log x}}$$

para x suficientemente grande (os logaritmos são todos na base e). Vale comparar este resultado com o Teorema dos Números Primos, conjecturado por Gauß e provado independentemente por Hadamard e de la Vallée Poussin em 1896, que diz que o número de primos até x é

$$\pi(x) \approx \frac{x}{\log x}.$$

Em particular, há muito mais primos do que pseudoprimos. A proposição abaixo exibe uma família infinita de pseudoprimos na base a (para qualquer $a > 1$ dado).

Proposição 6 *Sejam $a > 1$ e p primo tal que $p > 2$ e p não divide $a^2 - 1$. Então*

$$n = \frac{a^{2p} - 1}{a^2 - 1} = \frac{a^p - 1}{a - 1} \cdot \frac{a^p + 1}{a + 1}$$

é um pseudoprimo na base a.

Demonstração: Como $a \pm 1$ são inversíveis módulo p e $a^p \equiv a \pmod p$ pelo pequeno teorema de Fermat,

$$\frac{a^p - 1}{a - 1} \equiv \frac{a^p + 1}{a + 1} \equiv 1 \pmod p$$

e verifica-se facilmente que estes números são ímpares donde $n \equiv 1 \pmod{2p}$, ou $n = 2kp + 1$ para k inteiro. Assim, como $a^{2p} \equiv 1 \pmod n$ temos $a^n = a^{2kp+1} = (a^{2p})^k \cdot a \equiv a \pmod n$. □

Uma ideia natural para contornar a dificuldade criada pela existência de pseudoprimos é a de testar vários valores de a. É um fato interessante que existam alguns raros números compostos n, chamados *números de Carmichael*, com a propriedade de que se $0 < a < n$ e $\text{mdc}(a, n) = 1$ então $a^{n-1} \equiv 1 \pmod{n}$; os menores números de Carmichael são:

$$561 = 3{\cdot}11{\cdot}17,\ 1105 = 5{\cdot}13{\cdot}17,\ 1729 = 7{\cdot}13{\cdot}19,\ 2465 = 5{\cdot}17{\cdot}29,\ 2821 = 7{\cdot}13{\cdot}31;$$

não é difícil verificar que estes números satisfazem a definição: encorajamos o leitor a tentar. Foi demonstrado recentemente por Alford, Granville e Pomerance que se $CN(x)$ é a quantidade de números de Carmichael menores do que x então

$$CN(x) \geq x^{2/7}$$

para x suficientemente grande, o que implica na existência de infinitos números de Carmichael.

Podemos refinar o conceito de pseudoprimo para definir *pseudoprimos fortes* na base a. Para definir quando n é um pseudoprimo forte na base a inicialmente escrevemos $n-1 = 2^k \cdot b$, com b ímpar. Se $n > 2$ é primo deve existir um menor valor de j para o qual $(a^b)^{2^j} \equiv 1 \pmod{n}$ (observe que por Fermat $(a^b)^{2^k} \equiv 1 \pmod{n}$). Se $j = 0$ isto significa que $a^b \equiv 1 \pmod{n}$; caso contrário temos $(a^b)^{2^{j-1}} \equiv -1 \pmod{n}$ já que -1 é o único valor de x diferente de 1 (módulo n) para o qual $x^2 \equiv 1 \pmod{n}$. Assim, dizemos que n composto ímpar é um pseudoprimo forte na base a se *ou* $a^b \equiv 1 \pmod{n}$ *ou* existe $j' < k$ com $(a^b)^{2^{j'}} \equiv -1 \pmod{n}$. Claramente todo pseudoprimo forte na base a é um pseudoprimo na base a mas pseudoprimos fortes são mais raros do que pseudoprimos. Observe também que se calcularmos $a^{n-1} \bmod n$ pelo algoritmo sugerido no início desta seção então os números $(a^b)^{2^{k-j}} \bmod n$ serão calculados de qualquer forma (são os c_{N-j} daquele algoritmo); assim este novo teste tem custo computacional praticamente igual ao do primeiro teste.

Existem infinitos pseudoprimos fortes em qualquer base $a > 1$: Pomerance provou que, se $SP\pi_a(x)$ é o número de pseudoprimos fortes na base a menores ou iguais a x então

$$SP\pi_a(x) \geq e^{(\log x)^{5/14}}$$

para todo x suficientemente grande (ver [7]). Não existem "números de Carmichael fortes": para todo número composto ímpar n existe $0 < a < n$ com $\text{mdc}(a, n) = 1$ e tal que n *não* é um pseudoprimo forte na base a. Melhor ainda, os valores de a que servem de testemunha para a não-primalidade de n são sempre relativamente frequentes. Mais precisamente, seja

$$\alpha(n) = \frac{1}{\varphi(n)} |\, \{a \mid 0 < a < n, n \text{ é um pseudoprimo forte na base } a\}\, |.$$

Então para todo número composto ímpar $n > 9$ temos $\alpha(n) \leq 1/4$. A igualdade vale exatamente para os compostos n das seguintes formas:

$$n = p_1 p_2, \quad p_1, p_2 \text{ primos}, \quad p_1 \equiv 3 \pmod{4}, \quad p_2 = 2p_1 - 1;$$
$$n = p_1 p_2 p_3, \quad p_1, p_2, p_3 \text{ primos}, \ p_i \equiv 3 \pmod{4}, \ n \text{ número de Carmichael.}$$

Os menores exemplos de números compostos n da primeira forma para a qual $\alpha(n) = 1/4$ são $n = 15 = 3 \cdot 5$, $n = 91 = 7 \cdot 13$ e $n = 703 = 19 \cdot 37$. O menor exemplo de número composto da segunda forma é $n = 8911 = 7 \cdot 19 \cdot 67$. Sabe-se que existem menos do que $N^{(\frac{1}{2}+\epsilon)}$ números compostos n destas formas menores do que N; conjectura-se que o número de compostos n da primeira forma seja maior do que $N^{(\frac{1}{2}-\epsilon)}$. Na maioria dos casos $\alpha(n)$ é muito menor.

O resultado acima serve de base para o *algoritmo Miller-Rabin*, um exemplo de teste de primalidade *probabilístico*. Dado n, tomamos t valores de a ao acaso no intervalo $1 < a < n$ e verificamos para cada a se n passa no teste de primalidade forte na base a. Se n for ímpar composto, a probabilidade de que um dado a acuse a não-primalidade de a é maior do que $3/4$ (pelo teorema); assim, a probabilidade de que n escape a t testes é menor do que 4^{-t}. Este tipo de teste é extremamente útil em aplicações (como em criptografia) onde é importante criar primos relativamente grandes mas não existe a preocupação com demonstrações ou com perfeição absoluta.

4 Testes determinísticos para números especiais

Em outros contextos estamos interessados em *testes determinísticos*: dado um inteiro positivo n, queremos decidir, com certeza e em tempo curto (i.e., polinomial no número de dígitos), se n é primo ou composto.

Podemos modificar o algoritmo de Miller-Rabin para torná-lo determinístico testando todos os valores de a em um intervalo suficientemente grande: uma famosa generalização da hipótese de Riemann implica que o intervalo de 1 até $2(\log n)^2$ já é grande o bastante ([2]). Este algoritmo é rápido e geral, mas infelizmente depende de uma conjectura.

O seguinte resultado mostra que sob certas hipóteses as contas do teste de Miller-Rabin podem demonstrar a primalidade de n (com um único valor de a). De fato, muitos dos maiores primos conhecidos são desta forma.

Teorema 7 (Proth) *Seja $n > 1$ inteiro. Suponha que $n - 1 = 2^k b$ com $2^k > b$. Então n é primo se, e somente se, existe um inteiro a com $a^{(n-1)/2} \equiv -1 \pmod{n}$.*

Demonstração: Suponha que existe a como acima. Seja $q \mid n$, q primo. Temos $(a^b)^{2^{k-1}} \equiv -1 \pmod{q}$ donde $\operatorname{ord}_q a^b = 2^k$. Assim $2^k \mid q - 1$ e $q \geq 2^k$. Se n for composto temos

$$n \geq (2^k + 1)^2 > (2^k b + 1) + 2^{k+1} > n,$$

uma contradição.

Se n é primo, podemos tomar a qualquer não quadrado módulo n; ou seja, metade dos inteiros entre 1 e $n - 1$ serve como a. □

Dada a fatoração de $n - 1$ temos como generalizar este resultado para tentar demonstrar a primalidade de n.

Proposição 8 *Seja $n > 1$. Se para cada fator primo q de $n - 1$ existe um inteiro a_q tal que $a_q^{n-1} \equiv 1 \pmod{n}$ e $a_q^{(n-1)/q} \not\equiv 1 \pmod{n}$ então n é primo.*

Demonstração: Seja q^{k_q} a maior potência de q que divide $n-1$. A ordem de a_q módulo n é um múltiplo de q^{k_q}, donde $\varphi(n)$ é um múltiplo de q^{k_q}. Como isto vale para todo fator primo q de $n - 1$, $\varphi(n)$ é um múltiplo de $n - 1$ e n é primo. □

Proposição 9 (Pocklington) *Se $n - 1 = q^k R$ onde q é primo e existe um inteiro a tal que $a^{n-1} \equiv 1 \pmod{n}$ e $\operatorname{mdc}(a^{(n-1)/q} - 1, n) = 1$ então qualquer fator primo de n é côngruo a 1 módulo q^k.*

Demonstração: Se p é um fator primo de n então $a^{n-1} \equiv 1 \pmod{p}$ e p não divide $a^{(n-1)/q} - 1$, donde $\operatorname{ord}_p a$, a ordem de a módulo p, divide $n - 1$ mas não divide $(n - 1)/q$. Assim, $q^k \mid \operatorname{ord}_p a \mid p - 1$, donde $p \equiv 1 \pmod{q^k}$. □

Corolário 10 *Se $n-1 = FR$, com $F > R$ e para todo fator primo q de F existe $a > 1$ tal que $a^{n-1} \equiv 1 \pmod{n}$ e $\mathrm{mdc}(a^{(n-1)/q} - 1, n) = 1$ então n é primo.*

Demonstração: Seja q um fator primo de F e q^k a maior potência de q que divide F; pela proposição anterior, todo fator primo de n deve ser côngruo a 1 módulo q^k. Como isto vale para qualquer fator primo de F, segue que qualquer fator primo de n deve ser côngruo a 1 módulo F. Como $F > \sqrt{n}$, isto implica que n é primo. □

De fato, basta conhecer um conjunto de fatores primos cujo produto seja maior do que $(n-1)^{1/3}$ para, usando o resultado de Pocklington, tentar demonstrar a primalidade de n (o que deixamos como exercício).

A melhor maneira conhecida de gerar primos extremamente grandes é procurar *primos de Mersenne*, i.e., números da forma $2^p - 1$, p primo. Deixamos como exercício para o leitor verificar que se $2^n - 1$ é primo então n é primo. Os primeiros primos de Mersenne são 3, 7, 31 e 127, correspondentes a $p = 2, 3, 5, 7$; por outro lado, $2^{11} - 1 = 2047 = 23 \cdot 89$. São conhecidos hoje (abril de 2011) 47 primos de Mersenne e os 9 maiores primos conhecidos são todos desta forma: o maior deles corresponde a $p = 43\,112\,609$ e tem 12 978 189 algarismos; os outros oito correspondem aos seguintes valores de p: 42 643 801, 37 156 667, 32 582 657, 30 402 457, 25 964 951, 24 036 583, 20 996 011, 13 466 917. Conjectura-se que existam infinitos primos de Mersenne. Lembramos que um inteiro positivo é *perfeito* se ele for igual à soma de seus divisores positivos próprios. Os primeiros números perfeitos são 6 e 28; não é difícil verificar que os números perfeitos pares são exatamente os números da forma $2^{p-1}(2^p - 1)$, onde $2^p - 1$ é primo de Mersenne. Um dos poucos problemas em aberto da Matemática que foram formulados na antiguidade é decidir se existem números perfeitos ímpares.

O critério de Lucas-Lehmer permite determinar com facilidade se um número de Mersenne é primo. Dado um primo $p > 2$, queremos decidir se $n = 2^p - 1$ é primo. Sejam $S_0 = 4$, $S_1 = 4^2 - 2 = 14$, ..., $S_{k+1} = S_k^2 - 2$. Temos que n é primo se e somente se S_{p-2} é múltiplo de $2^p - 1$. Esta seqüência cresce muito rápido, mas basta fazer as contas módulo $2^p - 1$ (ver [3], [5]). O seguinte fato crucial para a demonstração pode ser verificado pelo leitor: para todo $k \geq 0$,

$$S_k = (2 + \sqrt{3})^{2^k} + (2 - \sqrt{3})^{2^k}.$$

Note que um número de Mersenne M_p é escrito na base 2 como $111\ldots111$, com p algarismos. Uma generalização natural seriam os números escritos como $111\ldots111$ em outra base, isto é, números da forma $(B^p - 1)/(B - 1)$, onde B é a base. É fácil ver que um tal número só pode ser primo se p for primo. No caso $B = 10$ estes números são conhecidos como *repunits*. Não se conhece um critério análogo ao de Lucas-Lehmer para testar a primalidade de números deste tipo quando $B > 2$. O maior primo conhecido desta forma é

$$\frac{28\,839^{8\,317} - 1}{28\,838},$$

que tem 37 090 algarismos. Os únicos repunits (comprovadamente) primos conhecidos são para $p = 2, 19, 23, 317, 1\,031$. Recentemente (entre 1999 e 2007), foram descobertos os seguintes valores de p para os quais os repunits correspondentes são *provavelmente* primos, i.e., passam por diversos testes probabilísticos de primalidade: 49 081, 86 453, 109 297 e 270 343. De acordo com os testes já realizados, qualquer outro repunit primo deve ter mais de 400 000 dígitos.

5 O teste determinístico de Agrawal, Kayal e Saxena

Permaneceu em aberto por muito tempo se existe um critério de primalidade determinístico, rápido e geral. Este problema for resolvido por Agrawal, Kayal e Saxena, que criaram um tal algoritmo, também baseado no teorema de Fermat. Suponhamos que x é uma variável, a um inteiro e p um número primo. Usando o binômio de Newton temos que

$$(x+a)^p = \sum_{j=0}^{p} \binom{p}{j} x^{p-j} a^j,$$

mas nos casos em que j é diferente de 1 e p, temos $p \mid \binom{p}{j}$ logo todos os termos intermediários desta expansão são divisíveis por p, assim

$$(x+a)^p \equiv x^p + a^p \equiv x^p + a \pmod{p}$$

onde na última igualdade usamos o teorema de Fermat (aqui $P_0 \equiv P_1 \pmod{N}$ significa que existe um polinômio S de coeficientes inteiros com $P_1 - P_0 = NS$, ou, equivalentemente, que para todo k os coeficientes de x^k em P_0 e P_1 são côngruos módulo n). Reciprocamente, se $(x+a)^N \equiv x^N + a \pmod{N}$ para algum inteiro a com $\text{mdc}(a,n) = 1$, vemos que N divide todos os coeficientes binomiais $\binom{N}{j}$ com $0 < j < N$. Se N fosse composto e q é um fator primo de N, então

$$\binom{N}{q} = \frac{N(N-1)\dots(N-q+1)}{q(q-1)\dots 1}$$

Vemos que os únicos termos que são múltiplos de q nesta expressão são o N no numerador e o q no denominador, assim se q^k é a maior potência de q que divide N, temos que $q^k \nmid \binom{N}{q}$, logo $N \nmid \binom{N}{q}$, absurdo. Assim, N é primo. Desta forma obtemos o seguinte critério de primalidade:

$$\begin{aligned} &N \text{ é primo } \iff (x+a)^N \equiv x^N + a \pmod{N}, \text{ para todo } a < N \\ \iff &(x+a)^N \equiv x^N + a \pmod{N}, \text{ para algum } a < N, \text{ com } \text{mdc}(a,N) = 1. \end{aligned}$$

Este critério, por enquanto, é ineficiente, porque temos que calcular todos os coeficiente de $(x+a)^N$ e mostrar que todos os coeficientes intermediários são divisíveis por N. Outra observação importante é que se os polinômios $(x+a)^N$ e $x^N + a$ são iguais módulo N, então eles deixam o mesmo resto módulo N quando divididos por qualquer polinômio mônico. Em particular, se dividimos por $x^r - 1$ temos que

$$N \text{ é primo } \implies \begin{array}{l} (x+a)^N \equiv x^N + a \pmod{x^r - 1, N} \\ \text{para todo } a < N, r \in \mathbb{N} \end{array}$$

(onde $P_0 \equiv P_1 \pmod{Q, N}$ significa que existem polinômios R, S com coeficientes inteiros tais que $P_1 - P_0 = QR + NS$).

O fato importante, mostrado por Agrawal, Kayal e Saxena, é que para garantir a primalidade de N só precisamos testar que esta congruência é válida para um valor especial de r (na versão original um r primo para o qual $r-1$ tem um fator primo $q \geq 4\sqrt{r}\log N$, o qual divide a ordem de n módulo r) que depende polinomialmente de $\log N$ e alguns poucos valores de a (veja [1], [3]).

1. Entrada $N > 1$.
2. Se $N = a^b$ com $b > 1$, retorna COMPOSTO.
3. Encontrar o menor r tal que $\text{ord}_r N > \frac{1}{2}(\log N)^2$.

4. Se $\mathrm{mdc}(a, N) > 1$ para algum primo $a \leq r$, retorna COMPOSTO.

5. Se $\sqrt{N} < r$, retorna PRIMO.

6. Para $a = 1$ até $\lfloor \sqrt{\varphi(r)/2} \log_2 N \rfloor$ faça: Se $(x + a)^N \not\equiv x^N + a \pmod{x^r - 1, N}$, retorna COMPOSTO;

7. Retorna PRIMO.

A outra pergunta proposta por Gauß permanece em aberto: se existe algoritmo rápido para *fatorar* inteiros. Um tal algoritmo teria grande relevância prática, comprometendo as formas mais populares de criptografia. Existe ainda a possibilidade de que não exista um algoritmo rápido, mas que ainda assim exista uma máquina (no sentido físico) capaz de fatorar inteiros rapidamente. De fato, a mecânica quântica parece permitir a construção de um *computador quântico* e Peter Shor encontrou um "algoritmo" que permite a um computador quântico fatorar inteiros em tempo polinomial [8].

Referências

[1] M. Agrawal, N. Kayal e N. Saxena, *PRIMES is in P*, Ann. of Math. (2) 160 (2004), no. 2, 781–793.

[2] E. Bach, *Explicit bounds for primality testing and related problems*, Math. of Comp. 55 (1990), 355–380.

[3] F. Brochero, C. G. Moreira, N. C. Saldanha, E. Tengan, *Teoria dos números: um passeio com primos e outros números familiares pelo mundo inteiro*, Projeto Euclides, IMPA, 2010.

[4] M. Cipolla, *Sui numeri composti P, che verificano la congruenza di Fermat* $a^{P-1} \equiv 1 \pmod{P}$, Annali di Matematica (3) 9, 1904, 139–160.

[5] D. H. Lehmer, *An extended theory of Lucas' functions*, Ann. Math. 31 (1930), 419–448. Reimpresso em *Selected Papers*, (ed. D. McCarthy), vol 1, Ch. Babbage Res. Center, St. Pierre, Manitoba, Canada, 11–48 (1981).

[6] W. H. Mills, *A prime representing function*, Bull. Amer. Math. Soc., 53 (1947), 604.

[7] C. Pomerance, *A new lower bound for the pseudoprimes counting function*, Illinois J. Math, 26, 1982, 4–9.

[8] P. W. Shor, *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*, SIAM J. Comput. 26 (5), 1484-1509 (1997).