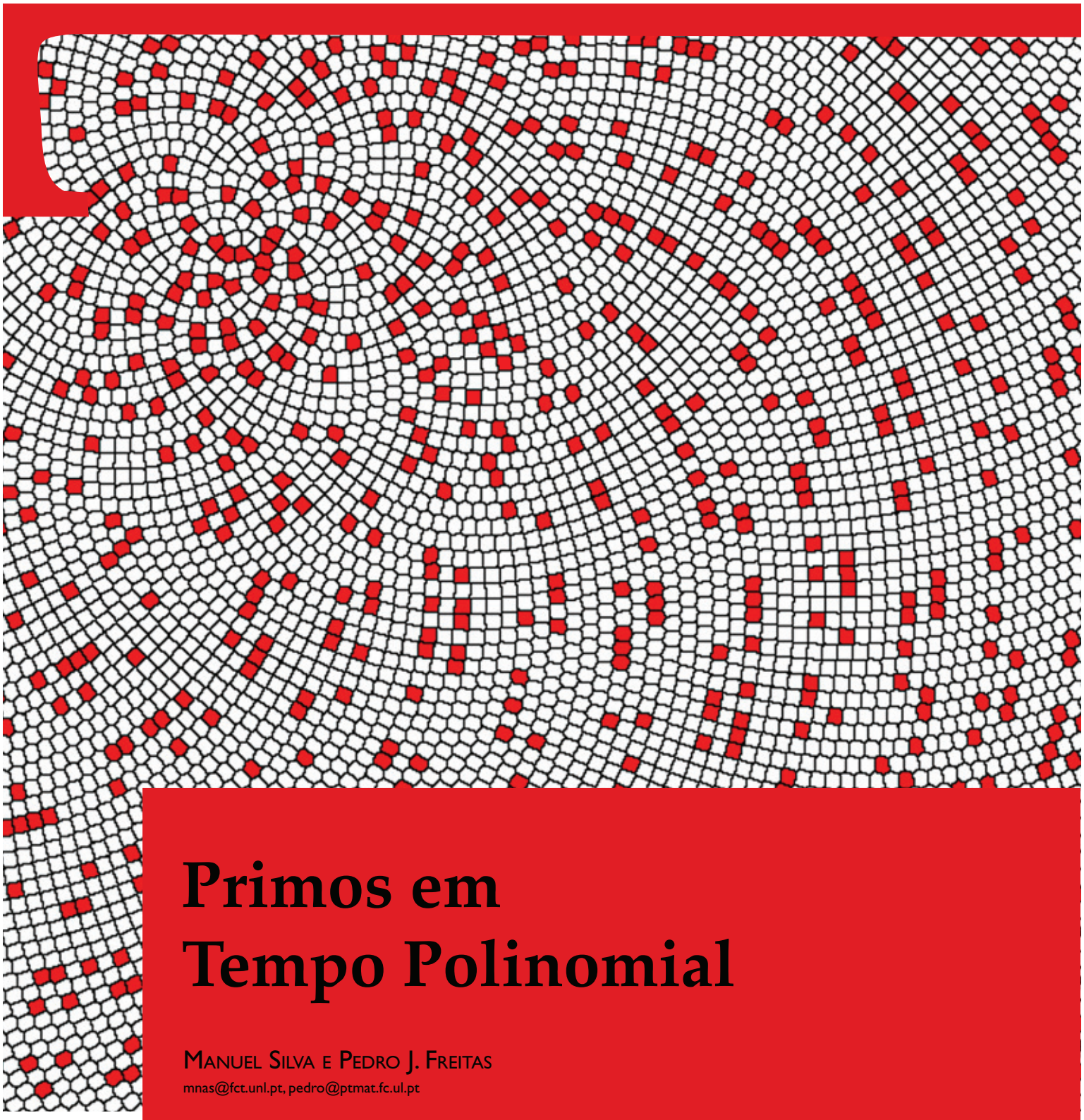


Primos em Tempo Polinomial

Manuel Silva e Pedro J. Freitas
mnas@fct.unl.pt, pedro@ptmat.fc.ul.pt

1. A IMPORTÂNCIA DOS PRIMOS

Não sabemos quem terá "inventado" os números primos, mas os gregos, já há cerca de 2300 anos, provaram por exemplo que os números primos nunca se esgotavam. Uma propriedade central destes números que faz deles objectos centrais da aritmética é o facto de todo o número natural se poder escrever de modo único como produto de primos, $120 = 2^3 \times 3 \times 5$ por exemplo.

Uma característica dos números primos que sempre fascinou os matemáticos e que os torna difíceis de estudar é a aparente imprevisibilidade na forma como se distribuem entre os números naturais. Por vezes parece que foram lá colocados de modo aleatório.

Deixemos os gregos e saltemos para o século XX. Será que o estudo dos primos tem alguma utilidade prática? Para desespero do matemático inglês Hardy, que preferia ser especialista numa área sem qualquer aplicação, os números primos revelam-se fundamentais em criptografia (codificação de mensagens). A utilidade dos primos em criptografia está relacionada com o facto de ser muito fácil multiplicar dois números primos a e b usando simplesmente o algoritmo que aprendemos na escola primária. No entanto, se me derem $n = ab$, eu terei em geral bastante dificuldade em descobrir os factores iniciais a e b, essenciais para descodificar a mensagem e os quais me foram ocultados.

O essencial dos métodos de criptografia com uma chave pública é a existência de operações difíceis de inverter. Multiplicar dois números *versus* factorizar no caso descrito.

Existem diversas maneiras de se saber um número é primo. A complexidade de um algoritmo (um conceito que se tornou relevante com a existência de computadores) mede o número mínimo de operações elementares necessárias para completar uma dada tarefa em função do tamanho dos dados iniciais.

Um algoritmo simples, que todos aprendemos, para verificar se um número inteiro é primo ou composto é dividi-lo por todos os inteiros até $\sqrt{n}$. Infelizmente, em termos computacionais, este algoritmo não é eficiente.

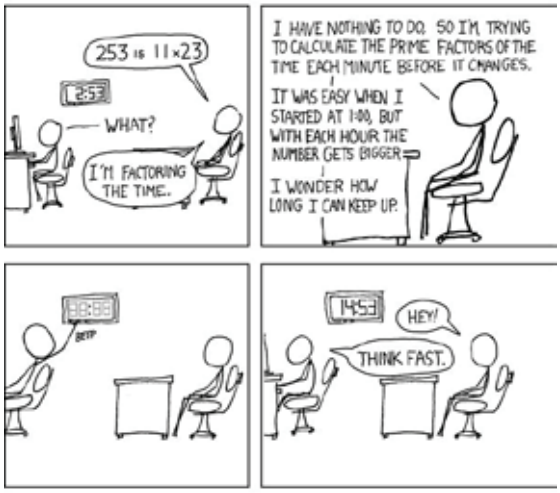
Considerando como dado inicial o número n, o seu tamanho é medido pelo seu número de dígitos 1, ou seja, $d = \log n$. Se precisarmos de dividir n por todos os números até $\sqrt{n} = 2^{\frac{d}{2}}$, o número de operações é exponencial em d, tamanho de n

"O problema de distinguir números primos de compostos, e de decompor números compostos nos seus fatores primos, é um dos mais importantes e úteis de toda a aritmética. (...) A dignidade da ciência parece pedir que nenhum esforço seja poupado na busca de uma solução para um problema tão famoso e elegante."

C. F. Gauss, *Disquisitionae Aritmeticae*, art. 329 (1798)

C. F. Gauss

medido como o número dos seus dígitos[1], o que essencialmente significa que, à medida que o número n cresce, a tarefa se torna rapidamente impossível de realizar, mesmo usando os computadores mais sofisticados.



Cartoon em *www.xkcd.com*

Assim, tentar factorizar um número para verificar se ele é ou não primo pode até funcionar se este tiver um factor pequeno, mas não é boa ideia em geral. Antes de continuar, notemos que encontrar uma factorização própria (ou provar que ela não existe) não é o mesmo que decidir se um número é primo. O que é que podemos fazer então?

Até 2002 existiam duas alternativas, ambas de algum modo insatisfatórias. Por um lado, eram conhecidos algoritmos determinísticos eficientes mas cuja justificação dependia de conjecturas em aberto. Por outro lado, existiam algoritmos probabilísticos eficientes, com uma probabilidade desprezável de o algoritmo certificar um número primo erradamente.

Em 2002, Agrawal, Kayal e Saxena, três cientistas da computação indianos (o segundo e o terceiro ainda alunos de licenciatura), descobriram, para espanto dos especialistas, um algoritmo que permite decidir se um número é primo em tempo polinomial.

O artigo onde este algoritmo é apresentado chama-se simplesmente "PRIMES is in P"[1], e afirma que o problema de determinar se um número é ou não primo, está em P, que é a classe dos problemas que se podem resolver computacionalmente em tempo polinomial.

Notamos aqui que, antes do artigo [1], o problema se encontrava em NP, a classe de problemas para os quais uma solução apresentada pode ser verificada em tempo polinomial. Por exemplo, se um número for composto, e nos derem uma factorização própria, é simples verificar se a factorização está certa, mas esta é difícil de encontrar. Se usamos o nosso cartão de crédito na Internet é porque ainda ningúem conseguiu encontrar um método simples para factorizar um número.

Um dos problemas matemáticos mais importantes, ainda em aberto, é justamente saber se para todo o problema na classe NP se pode encontrar uma solução algoritmica eficaz (i.e., em tempo polinomial), o famoso problema "P=NP". De um modo um pouco embaraçoso podemos dizer que não sabemos se existem problemas verdadeiramente difíceis, em termos computacionais! Esperemos que a resposta seja negativa, para podermos fazer compras na Internet e para continuarem a existir problemas matemáticos interessantes sem solução.

2. COMO RECONHECER UM PRIMO?

> "Os matemáticos tentaram em vão descobrir uma ordem subjacente à sequência dos primos, mas há razões para acreditar que existem mistérios onde nunca conseguiremos penetrar."
>
> L. Euler (1770)

Vamos agora descrever métodos que nos permitem decidir se um número inteiro n é primo sem ser necessário obter a sua factorização.

Aqui precisaremos da noção de congruência. Dois inteiros $a, b \in \mathbb{Z}$ são congruentes módulo n se $a - b$ for um múltiplo de n. Escrevemos $a \equiv b \mod n$, com um sinal parecido com o de igualdade porque de facto podemos trabalhar equações com congruências de forma similar às equações habituais, no que diz respeito a adições e multiplicações. Esta foi a razão pela qual Gauss o escolheu, no seu famoso *Disquisitiones Arithmeticae*, para representar a relação de congruência.

Se queremos então reconhecer eficazmente números primos, precisamos de encontrar uma propriedade no universo dos inteiros que seja verdadeira sempre que n é primo e só nesses casos. Além disso, a propriedade terá de ser fácil de testar (i.e., em tempo polinomial). Uma primeira tentativa seria a de tentar usar o pequeno teorema de Fermat: dado um primo n e um inteiro a, $a^n - a$ é sempre um múltiplo de n — isto é, se n é primo e a é um inteiro qualquer, $a^n \equiv a \mod n$. Veremos mais à frente uma demonstração simples deste resultado.

Assim, para garantir que um número n é composto basta encontrar um inteiro a (testemunha) tal que $a^n - a$ não seja múltiplo de n. Este método infelizmente não funciona sempre porque existem números que se "disfarçam" de primos, os denominados números de Carmichel: números compostos tais que para qualquer inteiro a, $a^n - a$ é sempre múltiplo de n. Estes números passariam em todos os testes relacionados com o pequeno teorema de Fermat mesmo não sendo primos.

Uma propriedade simples que, ao contrário do pequeno teorema de Fermat, caracteriza os primos de modo completo é dada pelo teorema de Wilson: um inteiro n é primo se e só $(n-1)!+1$ é um múltiplo de n — isto é, $(n-1)! \equiv -1 \mod n$. A dificuldade neste caso é não existir um modo simples de calcular $(n-1)!$[2].

Uma outra maneira de decidir se um número n é primo é olhar para a linha correspondente do Triângulo de Pascal e verificar se todos os elementos exceptuando os 1's das pontas são múltiplos de n. Podemos também escrever: n é primo se e só se $\binom{n}{k}$ é múltiplo de n para $1 \leq k \leq n-1$.

Este resultado não é muito difícil de demonstrar. Para verificar por exemplo que o coeficiente binomial $\binom{n}{k} = \frac{n(n-1)\cdots(n-k+1)}{k!}$ com $1 \leq k \leq n-1$ é múltiplo de n, basta observar que na fracção anterior o factor n aparece uma vez no denominador e nunca aparece no denominador porque n não tem factores próprios. O argumento para a implicação contrária é apenas um pouco mais envolvente, ficando ao cuidado do leitor dedicado. A dificuldade nesta caracterização consiste na necessidade de verificar $n-1$ coeficientes binomiais, o que é naturalmente pouco eficaz.

No entanto, esta caracterização tem uma formulação polinomial análoga que irá revelar-se crucial. Lembrando que

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^k y^{n-k},$$

se n for primo, temos $(x+y)^n = x^n + y^n + np(x,y)$, em que p é um polinómio, ou seja, $(x+y)^n \equiv x^n + y^n \mod n$, considerando agora congruências em polinómios. O artigo [4] chama a este resultado o *teorema binomial das crianças*.

Pode mesmo provar-se que n é primo se e só se $(x+1)^n \equiv x^n + 1 \mod n$, e a demonstração não é particularmente difícil.

Podemos apresentar nesta altura uma demonstração do pequeno teorema de Fermat, que afirma que para todo o inteiro a e todo o primo n, $a^n \equiv a \mod n$. A demonstração pode ser feita por indução: a propriedade é claramente válida para $a = 1$, e supondo que é válida para a, usamos o teorema binomial das crianças para escrever $(a+1)^n = a^n + 1^n \equiv a+1 \mod n$.

Usando o pequeno teorema de Fermat, vemos que se n é primo, então $(x+a)^n \equiv x^n + a \mod n$. Esta é uma das condições que aparecem no teorema, e que acabámos de provar ser necessária para n ser primo.

3. CARACTERIZAÇÃO DOS PRIMOS DE AGRAWAL, KAYAL E SAXENA

O resultado de Agrawal, Kayal e Saxena (AKS) baseia-se então numa caracterização engenhosa dos números primos, que permite decidir, em tempo polinomial, se um número é primo. Começamos por apresentar a caracterização, passando depois a esclarecer alguns conceitos.

Teorema *Seja $n \geq 2$ um inteiro, $r < n$ um inteiro positivo tal que n tem, módulo r, ordem maior do que $(\log n)^2$. Então n é primo se e só se*

- *n não é uma potência perfeita,*
- *n não tem nenhum factor primo menor ou igual a r e*
- *para cada inteiro a, $1 \leq a \leq \sqrt{r}\log n$, temos $(x+a)^n \equiv x^n + a \mod (n, x^r - 1)$.*

Esclarecemos agora alguns conceitos: a *ordem* de um inteiro $m \mod r$, para m e r coprimos, é o menor número k tal que $m^k \equiv 1 \mod r$. Na terceira condição, a notação $\mod (n, x^r - 1)$ indica congruência módulo n e módulo $x^r - 1$ simultaneamente. Encontra-se uma caracterização equivalente em [3], apresentada inicialmente por Dan Bernstein em [2].

Destas condições, já verificámos que algumas são necessárias para a primalidade de n. A demonstração completa da caracterização, embora curta, não é do âmbito deste artigo, e pode encontrar-se em [1], [4] e [3].

Não vamos igualmente demonstrar que o tempo computacional de implementação desta caracterização é polinomial

[1] Em questões computacionais é mais habitual usar a base 2 em vez da base 10 que costumamos usar na escrita de números.

[2] É talvez surpreendente que não se conheça uma forma simples de multiplicar os primeiros n naturais análoga à que conhecemos para os somar.

(mais uma vez, remetemos o leitor interessado para os artigos acima). Para dar uma ideia da simplicidade de alguns dos cálculos necessários, como o cálculo da potência de $(x+a)^n$, note-se que à medida que vamos fazendo multiplicações, como estamos a fazê-las no módulo $x^r - 1$, o grau dos polinómios que aparecem nunca é maior do que r. O resultado inicial de AKS afirma que o algoritmo termina ao fim de um tempo na ordem de $d^{7.5}$, em que $d = \log_2 n$ corresponde ao número de dígitos de n em base 2.

4. COMENTÁRIOS FINAIS

O resultado de AKS foi notável a vários níveis. Em primeiro lugar, pela sua simplicidade: o artigo inicial que deu origem a [1] tem nove páginas, e os especialistas a quem o artigo tinha sido enviado confirmaram em poucos dias que o argumento estava essencialmente correcto. Foi em 4 de Agosto de 2002, um domingo, que os autores enviaram a primeira versão do artigo a 15 especialistas. Nessa mesma noite receberam já algumas respostas de parabéns, na segunda-feira Carl Pomerance (um dos 15 especialistas) organizou um seminário improvisado para discutir o assunto e preveniu o *The New York Times*, que publicou na quinta-feira seguinte uma notícia sobre o resultado, com título "New Method Said to Solve Key Problem in Math". Na sexta-feira já estava publicada na internet uma versão abreviada da demonstração, que se reduzia a uma página (da qual [2] contém uma versão posterior). Outras contribuições foram dadas mais tarde, como por exemplo a de Henrik Lenstra e Carl Pomerance, que apresentaram em 2003 uma modificação do algoritmo que funcionava em tempo da ordem de d^6.

É muito invulgar que um problema tão antigo e importante seja resolvido de forma tão simples. O artigo [3] nota que é praticamente impossível explicar a um grande público, mesmo com alguma formação matemática, soluções completas de outros problemas antigos, como o último teorema de Fermat, ou o trabalho dos últimos laureados com a medalha Fields. No entanto, é notável que esta solução seja compreensível até para alunos de licenciatura.

Quase a terminar temos de confessar que o algoritmo aqui proposto por AKS, embora de complexidade polinomial e sob diversos aspectos, brilhantes, não foi usado na prática, pelo menos, até hoje. Na verdade, os algoritmos mais eficazes para determinar se um inteiro n é primo são os algoritmos probabilísticos referidos anteriormente. Embora exista a possibilidade destes algoritmos certificarem como primos números inteiros n erradamente, isto acontece com uma probabilidade tão pequena quanto se queira – inferior a 10^{-50} por exemplo –, mais facilmente alguém ganharia 10 vezes consecutivas a lotaria.

Finalmente resta dizer que desde 1997 existem algoritmos quânticos para factorizar números inteiros em tempo polinomial. Mas o leitor pode estar descansado e continuar a usar o seu cartão de crédito porque o algoritmo descoberto por Peter Shor [5] depende da construção de uma nova geração, ainda inexistente, de computadores que procuram tirar partido de fenómenos quânticos.

REFERÊNCIAS

[1] M Agrawal, N Kayal, N Saxena, "PRIMES is in P", *Annals of Mathematics*, 160 (2004), 781-793.

[2] Dan Bernstein, "Proving Primality after Agrawal-Kayal-Saxena", version of January 25, 2003, *http://cr.yp.to/papers.html#aks*

[3] Folkmar Bornemann, "PRIMES Is in P: A Breakthrough for 'Everyman'", *Notices of the AMS*, Maio de 2003.

[4] Andrew Granville, "It is Easy to Determine Whether a Given Integer is Prime", *Bulletin of the AMS* 42 (2005), 3-38 (2008 Chauvenet Prize).

[5] Peter Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer", SIAM J. Comput. 26 (5), 1484–1509.

SOBRE OS AUTORES

Manuel Silva obteve em 2008 o grau de Doutor em Matemática pela Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa. Faz investigação em combinatória e teoria aditiva de números. Actualmente é professor no Departamento de Matemática da FCT da Universidade Nova de Lisboa e membro do CMA - Centro de Matemática e Aplicações

Pedro J. Freitas obteve o grau de Doutor (PhD) em Matemática pela Universidade do Illinois em Chicago. Actualmente é professor no Departamento de Matemática da Universidade de Lisboa. Desenvolve investigação nas áreas de análise matricial e teoria da representação.