Testes de Primalidade

Gervásio Protásio dos Santos Neto[*†]

26 de janeiro de 2015

Resumo

O principal objetivo deste projeto descrever os fundamentos matemáticos, implementação e uso de alguns algoritmos importantes para Criptografia e Teoria dos Números: os testes de primalidade. Para cada teste estudamos os resultados matemáticos que o fundamentam e o algoritmo em si. Ao final apresentamos os resultados obtidos nos testes e uma pequena comparação das performances.

Palavras-chave: criptografia, números primos, algoritmos.

Abstract

The main goal of this project was to describe the mathematical fundamentals, the implementation and use of a class of algorithms that is important for Cryptography and Number Theory: primality tests. For each selected test we studied the its mathematical basis, the algorithm and its performance. In the end, we compared the results obtained from the tests.

Keywords: cryptography, prime numbers, algorithms.

1 Introdução

Alguns dos resultados mais importantes de Teoria dos Números, no que diz respeito à Criptografia, se relacionam com números primos. Esse conjunto infinito de números possuem diversas aplicações na área. Por exemplo, são fundamentais para o RSA e para Criptossistemas de Curvas Elípticas (CCEs), ambos amplamente utilizados para encriptação de dados e dependendentes de primos para funcionarem corretamente e proverem algum tipo de segurança.

Então, dada a importância que números primos têm, é natural que busque-se formas rápidas e eficientes de encontrá-los. Mais especificamente, é interessante que, dado um número, sejamos capazes de rapidamente determinar se trata-se ou não de um número primo. Com tal finalidade foram desenvolvidos diversos algoritmos (testes de primalidade) que, fundamentando-se em resultados da Teoria dos Números, são capazes de determinar se um número é primo (ou se o é com alta probabilidade).

Este artigo procura descrever o estudo e implementações e testes de diversos testes de primalidade. No seu restante, descreveremos as formas de implementação dos testes, os resultados obtidos, os problemas encontrados durante o desenvolvimento e e as conclusões obtidas ao longo do processo.

2 Objetivos

Neste trabalho buscamos:

- Estudar a fundamentação teórica que possibilita alguns testes importatantes;
- Implementá-los para chegar a algumas conclusões sobre as reais eficácias e aplicabilidades do teste;
- Quando relevante, tentamos propor novas ideias, heurísticas ou formas de implementação que possibilitem seu melhoramento

Os testes que escolhidos o foram por corresponder a algum dos seguintes critérios: relevância teórica, aplicabilidade prática, importância histórica.

Ignorou-se o teste trivial de divisões sucessivas e começou-se pelo teste de Wilson, que foi escolhido pois o teorema no qual se fundamenta é um resultado clássico de Aritmética Modular.

[*]e-mail: gervasio.neto@usp.br

[†]Com bolsa de Iniciação Científica do Conselho Nacional de Desenvolvimento Científico e Tecnológico.

Um grande foco foi dado para o entendimento de um teste probabilístico, o teste de Miller-Rabin [1], baseado na inversão do Pequeno Teorema de Fermat. Sua correção é demonstrada por resultados profundos de Teoria dos Números e o algoritmo mostra-se eficiente.

O Teste AKS [4] também foi estudado por ter grande importância teórica, sendo o primeiro teste de primalidade determinístico e em tempo polinomial no número de bits do número (ou seja, polinomial em $O(\log n)$. Sua implementação foi feita e surtiu resultados interessantes.

Por fim, buscamos avaliar a aplicabilidade de curvas elípticas a testes de primalidade, principalmente o protocolo de autenticação proposto por Goldwasser e Kilian [5].

3 Materiais e Métodos

O estudo foi inicialmente desenvolvido da seguinte forma:

Primeiro, foram selecionados os testes a serem estudados. Foram escolhidos:

- Teste de Wilson [2], por ser inspirado no clássico Teorema de Wilson da Artimética Modulars;
- Teste de Miller-Rabin [1][2], por ser amplamente utilizado, dado sua eficiência e ser um exemplo clássico de um algortimo que troca determinismo por eficiência;
- Algoritmo AKS [4], devido a sua relevância teórica, uma vez que foi responsável por mostrar que é possível decidir primalidade em tempo polinomial
- Teste de Goldwasser-Kilian [3], por ser a primeira proposta de uso de curvas elípticas em testes de primalidade.

Uma vez que os testes a serem estudados foram selecionados, buscou-se entender os principais teoremas que os fundamentam e provar resultados importantes relacionados à aplicabilidade deles.

Com a fundamentação matemática completa, buscou-se implementar alguns destes testes como programas executáveis e medir empiricamente sua eficiência. Inicialmente escolheu-se fazê-lo em linguagem C, dado o baixo *overhead* dela. Contudo, para estudar o teste de Goldwasser-Killian, utilizou-se a linguagem SAGE; os outros testes foram então traduzidos para essa liguagem e testados para fins de comparação. Os tempos de desempenho aqui apresentado correspodem aos tempos das implementações em SAGE.

Todos os teste foram realizados em um notebook MacBook Air, 1.7 GHz Intel Core i5, memória de 4GB 1333 MHz DDR3, sistema operacional OS X Mavericks.

Por fim, reuniram-se conclusões sobre eficiência e aplicabilidade dos algoritmos, problemas e dificuldades de implementação e propõe-se ideias que poderiam levar a sua melhoria.

Vale mencionar que um teste utilizado para melhor entender a riqueza do campo, mas não estudado com profundidade foi o Teste de Lucas-Lehmer para primos de Mersenne. Tal teste é o utilizado pelo projeto *Great Internet Mersenne Prime Search* para achar novos primos de Mersenne e foi o responsável por achar o maior número primo conhecido atualmente.

4 Resultados

O teste de Wilson (implementado em C e SAGE) mostrou-se lento. Isso se deve ao cálculo do fatorial, que faz com que o algoritmo seja exponencial no tamanho da entrada. Isso significa que o teste é inviável para aplicações práticas

O teste de Miller-Rabin foi implementado em C e em SAGE e mostrou-se eficiente. Em C, para primos da ordem de 10^9 (tamanho do maior inteiro possível em C) o tempo médio foi em torno de 61,7 *ns*, com probabilidade de erro de 4^{-30}.

Em ambas as implementações os tempos para números compostos foram negligenciáveis. Percebeu-se que o algoritmo determinava rapidamente se um número era composto e demorava-se apenas para primos (uma vez que nestes casos, o algoritmo não era interrompido, tendo todos seus passos realizados).

Para o AKS, verificou-se que uma implementação em C do pseudo-código era ineficiente e de difícil implementação. Tentou-se algumas mudanças na busca por melhorias:

- Procurou-se um limite superior diferente para o laço do do algoritmo, pois originalmente é necessário o cálculo de $\varphi(n)$, uma atividade computacionalmente custosa.

- Ao invés de verificarmos uma desigualdade de polinômios, sorteamos um número, calculamos as funções e vemos se os valores são condizentes.

Verificou-se que a primeira mudança não nos trouxe ganho significativo de performance, enquanto a segunda o fez, mas abrindo mão do determinísmo. Conclui-se que o teste é de interesse apena teórico, por ser responsável por mostrar que existe um algoritmo polinomial determinístico para decidir primalidade. A versão em SAGE, feita para fins de comparação é uma tradução da versão em C e tem desempenho similar.

Quanto ao algoritmo de Goldwasser-Killian, para testar a eficiência do algoritmo usou-se uma implementação ufeita por Georg Hahn e presente em `http://trac.sagemath.org/attachment/ticket/10562/ecpp.py`, sendo esse implementação distribuida sob licença GNU Public License (GPL).

O algoritmo mostrou-se, em geral, satisfatoriamente rápido. Foram rodados testes para verificar o tempo médio necessário para decidir a primalidade de um número (as entradas eram números primos aleatórios) e para verificar o certificado de primalidade. Para números da ordem de 10^{30} observou-se em média 2.56 segundos para obter o certificado de primalidade; caso já se tivesse um certificado, este era testado com tempo médio de 0.078 segundos. Com entradas da ordem de 10^{40} observou-se um tempo médio de 20.39 segundos para obtenção de um certificado e 0.448 segundos para a verificação de um certificado .

Com base nestes números, somos levados a concluir que, apesar de não ser tão eficiente quanto testes mais estabelecidos de primalidade (como o de Miller-Rabin), o teste de Goldwasser-Kilian ainda consegue ser satisfatoriamente rápido. Além disso, é um teste determinístico que nos dá um certificado de primalidade que pode ser testado rapidamente, ambas vantagens sobre o teste de Miller-Rabin.

O que temos, do ponto de vista teórico, é que o algoritmo é penalizado na etapa 3 (a contagem de pontos). Apesar de conhecer-se o Algoritmo de Schoof ([12]) para contar os pontos de uma curva sobre um corpo finito, tal algoritmo possui difícil implementação e baseia-se em resultados profundos de Teoria dos Números e curvas elípticas. Além disso, implementações eficientes, mesmo que polinomiais, ainda possuem consumo de tempo de $O((\log n)^5)$. Algoritmos anteriores ao de Schoof tinham consumo de tempo exponecial, portanto, tornam-se impráticos para validar a primalidade de números relevantes.

Na tabela a seguir encontram-se os tempos (em segundos) dos algoritmos (em suas implementações em SAGE) para primos de diversas ordens de grandeza. As entradas foram obtidas usando a função nativa do SAGE *random_prime* , que recebe com argumento uma ordem de grandeza e devolve um primo aleatório desta magnitude. Entradas marcadas com '-' representam tempo cuja medida foi impraticável.

Algoritmo	10^4	10^5	10^6	10^7	10^8	10^9	10^{10}	10^{30}	10^{40}
Wilson	0.0022	0.0221	0.151	2.226	20.5	-	-	-	-
Miller-Rabin	0.0015	0.0018	0.0020	0.0023	0.0025	0.0027	0.0029	0.0078	0.0414
AKS	0.085	0.182	0.343	0.487	1.03	1.412	2.292	-	-
Goldwasser-Kilian	2.6e-5	2.71e-5	2.74e-5	0.0095	0.0152	0.0218	0.0317	2.56	20.39

Tabela 1: Tabela com os tempos (em segundos) dos algoritmos para primos em diversas ordens de grandeza

5 Conclusões

Baseando-nos nos números coletados, somos levados a concluir que os teste de Wilson e AKS são relevantes apenas teoricamente, sendo pouco interessantes em um contexto aplicado.

O teste de Miller-Rabin mostra-se uma excelente forma de decidir a primalidade de um número, não tendo em sua inerente aleatoriedade uma desvantagem (uma vez que podemos estabelecer uma precisão arbitrária para nossos resultados).

O teste de Goldwasser-Kilian aparenta ser satisfatoriamente rápido. Apesar de mais lento que o algoritmo de Miller-Rabin, este algoritmos é determinístic e nos dá um certificado de primalidade que pode ser testado rapidamente, como previsto por Goldwasser e Kilian [5][11] e verificado por Atkin e Morain [8].

6 Bibliografia

Referências

[1] Routo Terada, *Segurança de Dados*. Blucher, São Paulo, Segunda Edição, 2008.

[2] E. Kranakis. *Primality and Criptography*. Wiley, 1986.

[3] N. Koblitz. *A Course in Number Theory and Criptography*. Springer-Verlag, 1994.

[4] M. Agrawal, N. Kayal, N. Saxena. *PRIMES is in P. Annals of Mathematics*, 160 (2):781–793, 2002

[5] S. Goldwasser, J. Kilian. *Almost all primes can be quickly certified. Proc. 18th Annual ACM Symposium on Theory of Computing*, :316–329, 1986.

[6] François Morain. *Pseudoprimes: A Survey Of Recent Results*, *Proc. Eurocode '92* , Springer (1993), pp. 207–215. 1993.

[7] W. R. Alford and Andrew Granville and Carl Pomerance. *There are infinitely many Carmichael numbers*, *ANN. OF MATH*, 1982, 140, 703–722.

[8] A. O. L. Atkin and F. Morain. *Elliptic Curves And Primality Proving*, *AMS Mathematics of Computation*, 61, 29–68, 1993.

[9] Rafael Dantas de Castro, Ricardo Dahab e Augusto Jun Devegili. *Introdução à Segurança Demonstrável. VII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais* , SBC, 103 – 153. 2007.

[10] H. Lenstra, C. Pomerance. Primality testing with gaussian periods. Private communication. Março, 2003.

[11] S. Goldwasser, J. Kilian. *Primality Testing Using Elliptic Curves. Journal of the ACM*, Vol 46, No. 4, Julho 1999, pp. 450-472.

[12] Gregg Musiker. Schoof's Algorithm for Counting Points on $E(\mathbb{F}_q)$. Dezembro, 2005.

[13] Washington, Lawrence C. *Elliptic Curves: Number Theory and Criptography Chapman & Hall/CRC, 2nd Edition, 2008*