TEORIA E PRÁTICA NA BUSCA DE NÚMEROS PRIMOS DE MERSENNE

Comissão Técnica: Prof. Dr. Edival de Morais
Prof. M. Sc. Eduardo Quadros da Silva
Profa. Dra. Maria da Conceição Pinheiro
Autores: Prof. M. Sc. Leonardo Brodbeck Chaves
Paulo Amaro Velloso H dos Santos
Centro Universitário Campos de Andrade – Uniandrade
Departamento de Matemática
Campus Deodoro – Centro
Av. Marechal Deodoro, 1028 – CEP 80010-010
lbchaves@suisse.net / paulo@suisse.net

RESUMO

Marin Mersenne (1588-1648) foi um frei franciscano francês que dedicou boa parte de sua vida ao estudo da matemática, escrevendo livros e contribuindo para o desenvolvimento de teorias matemáticas através de correspondências com Fermat e outros matemáticos da época. Estudaremos com mais detalhes a teoria ligada aos números primos de Mersenne, que está intimamente ligada aos 'números perfeitos', que são obtidos através da fórmula $n = 2^{m-1} \times (2^m - 1)$, onde m é um número inteiro positivo e o fator $(2^m - 1)$ é um número primo. Ao número primo da forma $(2^m - 1)$ chamamos de Número Primo de Mersenne. Os primeiros primos de Mersenne são 3, 7, 31, 127. Como esses números estão relacionados a uma função exponencial, seu crescimento é acelerado, atingido rapidamente centenas de dígitos. Curiosamente, até esta data só foram encontrados 39 primos de Mersenne, sendo que o maior deles contém 4.053.946 dígitos. Neste trabalho, apresentaremos a teoria matemática e alguns algoritmos que podem ser utilizados na busca desses números. Além disso, serão apontadas algumas diretrizes para a construção de um programa de computador, baseadas nessa teoria e algoritmos, que possa servir na busca de números primos de Mersenne ainda desconhecidos.

1 INTRODUÇÃO

O conceito de número primo de Mersenne está relacionado ao conceito de "número perfeito". Um número perfeito é um inteiro no qual a soma de seus divisores é o dobro do número. Por exemplo,

$$6 = 1 + 2 + 3 + 6 = 12 = 2 \times 6$$

Dessa maneira, 6 é um número perfeito.

Os gregos descobriram que todo número perfeito é da forma $n = 2^{m-1} \times (2^m - 1)$, onde *m* é um inteiro com $m \geq 2$, e $2^m - 1$ é primo.

Isto significa que a busca de números perfeitos é reduzida à busca de números primos da forma $2^m - 1$.

O número da forma $2^m - 1$é chamado de número de Mersenne. Um número primo dessa forma é chamado de número primo de Mersenne.

2 CONTEXTO HISTÓRICO

Marin Mersenne (1588-1648) foi frei franciscano francês que viveu a maior parte de sua vida em um mosteiro de Paris. Foi autor de *"Cognitata Physico-Mathematica"* no qual afirmou, sem prova, que $2^m - 1$ é primo para m igual a 2, 3, 5, 7, 13, 17, 19, 31, 67, 127 e 257 e para nenhum outro primo *m*, para *m < 257*.

Um trabalho feito em 1947 mostrou que Mersenne cometeu cinco erros em seu trabalho (2^{61} é primo, $2^{47} - 1$ é composto, 2^{89} é primo, $2^{107} - 1$ é primo e $2^{257} - 1$ é composto). Além de sua famosa afirmação sobre números primos da forma $2^m - 1$, Mersenne contribuiu para o desenvolvimento da teoria dos números através de sua vasta correspondência com vários matemáticos da época, incluindo Fermat. Mersenne efetivamente serviu como um investigador e disseminador de novas idéias matemáticas do século XVII.

3 A BUSCA DE NÚMEROS PRIMOS DE MERSENNE

Como já visto, números primos de Mersenne são números primos da forma $2^m - 1$, onde m é um inteiro sendo $m \geq 2$. Os primeiros números primos de Mersenne são 3, 7, 31 e 127. O curioso é que até a data da escrita deste artigo, apenas 39 números primos de Mersenne são conhecidos. Na prática, os primos de Mersenne crescem aceleradamente, formando rapidamente inteiros com centenas de dígitos. Isso faz com que o custo computacional, ou o tempo gasto na busca, torne-se demasiadamente alto. A saída é possuir um hardware o mais poderoso possível, conjugado a algoritmos otimizados para essa busca.

A seguir é apresentada a teoria Matemática a partir da qual surgem algoritmos computacionais que podem ser utilizados na busca dos números primos de Mersenne. Em seguida, são apresentados os números de Mersenne encontrados através desses algoritmos, através de um programa de computador escrito em linguagem C++.

4 TEORIA MATEMÁTICA E ALGORITMOS ÚTEIS NA BUSCA DOS NÚMEROS PRIMOS DE MERSENNE

4.1 Teoremas

Teorema 1: Se *m* é um inteiro positivo e $2^m - 1$ é primo, então *m* também é primo.

Prova: Sejam *r* e *s* inteiros positivos. O polinômio $x^{rs} - 1$ pode ser escrito como:

$$x^{rs} - 1 = (x^s - 1).(x^{s(r-1)} + x^{s(r-2)} + x^{s(r-3)} + ... + x^s + 1)$$

Se *m* é composto, com $m = rs$, e $1 < s < m$, então $2^m - 1$ também é composto, pois é divisível por $2^s - 1$.

Teorema 2: Se *m* é um número primo, então qualquer divisor do número primo de Mersenne $2^m - 1$ é da forma $2.k.p + 1$, onde *k* é um inteiro positivo.

Prova: Se *p* é divisor de $2^q - 1$, então $2^q = 1 (\mathrm{mod}\ p)$ e a ordem de $2 (\mathrm{mod}\ p)$ é divisor do primo *q*, então deve ser *q*. Pelo Pequeno Teorema de Fermat, a ordem de 2 também divide $p - 1$, então $p - 1 = 2.k.q$.

Assim,

$$2^{\frac{(p-1)}{2}} = 2^{k.q} = 1 (\mathrm{mod}\ p)$$

Então 2 é um resíduo quadrático $\mathrm{mod}\ p$ e segue que $p = \pm 1 (\mathrm{mod}\, 8)$, completando a prova.

Teorema 3: Pequeno Teorema de Fermat.
Seja *p* um primo que não é divisor do inteiro a, então $a^{p-1} = 1 (\mathrm{mod}\ p)$.

Prova: Sejam os $p - 1$ primeiros múltiplos positivos de a:

$$a, 2a, 3a, ..., (p-1).a$$

Suponha que $r.a$ e $s.a$ são o mesmo módulo p, então temos $r = s (\mathrm{mod}\ p)$, então os $p - 1$ múltiplos de a acima são distintos e deferentes de zero; ou seja, devem ser congruentes a 1, 2, 3,....., $p - 1$ e alguma ordem. Multiplique todas essas congruências e encontramos

$$a.2a.3a.....(p-1)a = 1.2.3.....(p-1) (\mathrm{mod}\ p),$$

ou melhor,

$$a^{(p-1)}.(p-1)! = (p-1)! (\mathrm{mod}\ p).$$

Quando dividimos ambos os membros por $(p-1)!$ temos a prova completa.

4.2 Algoritmos

Peneira de Eratosthenes

A Peneira de Erathostenes nada mais é que um algoritmo para formar uma tabela de números primos.

O funcionamento do algoritmo é o seguinte: seqüencialmente escrever os números inteiros desde 2 até o maior número *n* que se queira incluir na tabela. Marque com um risco todos os números maiores que 2 que são divisíveis por 2 (a cada segundo número). Encontre o menor número remanescente maior que (3). Marque com um risco os números maiores que 3 que são divisíveis por 3 (a cada terceiro número). Ache o menor número remanescente maior que 3 (5). Marque todos os números maiores que 5 que são divisíveis por 5 (a cada quinto número).

Continuar até que se tenham marcado todos os números que são divisíveis por $\left\lfloor \sqrt{n} \right\rfloor$, onde $[x]$ é a função que retorna o menor inteiro mais próximo de *x*. Os números restantes (não marcados) são primos. Este procedimento é ilustrado na figura abaixo, para determinação dos primos até o número 50. Note que são marcados os números até $\left\lfloor \sqrt{50} \right\rfloor = 7$.

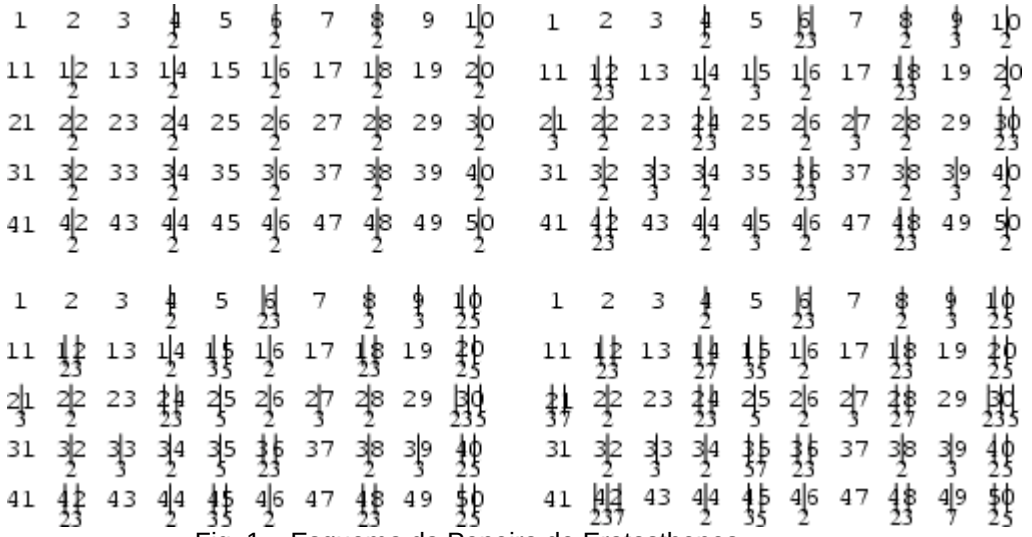

Fig. 1 – Esquema da Peneira de Eratosthenes

Teste de Lucas – Lehmer

Este teste é baseado no seguinte critério: sejam $S_0 = 4, S_1 = 4^2\text{-}2 = 14, ..., S_{k+1} = S_k{}^2\text{-}2$; dado $p > 2$, $2^p - 1$ é primo se e somente se S_{p-2} é múltiplo de $2^p - 1$.

Por motivo de espaço, omitiremos o prova deste critério. A seguir está exemplificada uma possível implementação desse teste na linguagem C++.

```
R = 4;
for (i = 1; i ≤ p - 2; i++)
{
    R *= R;
    R -= 2;
    R = R % Mm
}
if (R == 0)
  //Mm é primo
```

Na prática a complexidade deste algoritmo é limitado por $O(n^2)$, ou seja, o custo da multiplicação $R * R$.

5 UTILIZAÇÃO PRÁTICA DOS TEOREMAS E ALGORITMOS

Foram implementados dois programas, na linguagem C++, utilizando a teoria e os algoritmos discutidos na seção anterior.

A busca de um número primo da forma $2^m - 1$ consiste, em termos computacionais, ao teste da primalidade de $2^m - 1$. Esse teste pode ser otimizado através do uso do Teorema 1, pois o teste de primalidade de $2^m - 1$ estaria limitado aos testes dos casos em que *m* é primo. O teste pode ser ainda mais otimizado utilizando-se o Teorema 2, já que os candidatos a fator do número $2^m - 1$ são da forma $2.k.p + 1$, onde *k* é um número inteiro e positivo.

Esses dois teoremas da Seção 4.1, em conjunto com a peneira de Eratosthenes, aliados ao conhecimento de que um fator primo de $2^m - 1$ deve ser menor que $\sqrt{2^m - 1}$, são suficientes para viabilizar computacionalmente a determinação da primalidade dos números de Mersenne para $m < 257$.

Para validar essas afirmações, foi implementado o programa denominado *Mersenne v.1*, que utiliza-se apenas dos teoremas 1 e 2 da seção 4.1 e do algoritmo resultante de Peneira de Eratosthenes, da Seção 4.2.

O segundo programa, denominado *Mersenne v.2*, utiliza, além da teoria e do algoritmo de Eratosthenes, o Teste de Lucas – Lehmer. O segundo programa mostra-se mais eficiente, porém limitado basicamente ao tempo de multiplicação de grandes inteiros do computador utilizado.

Os resultados dos programas *Mersenne v.1 e Mersenne v.2* até a data da finalização deste artigo, ainda estão sendo melhor avaliados, pois os programas ainda estão em fase alfa de desenvolvimento. Para obter o estágio de implementação atual, bem como o código-fonte, contate via e-mail um dos autores.

A linguagem escolhida foi a linguagem C++ através do compilador gcc, funcionando no sistema operacional Linux (www.linux.org), em conjunto com a

biblioteca de precisão aritmética arbitrária GMP (http://swox.com/gmp) . Todos os softwares utilizados na construção e teste dos programas são de livre utilização e distribuição.

6 CONCLUSÕES E TRABALHOS FUTUROS

Apesar de até a data atual o programa *Mersenne v.1* estar em sua fase alfa de desenvolvimento, a teoria utilizada, mesmo que aplicadas as devidas otimizações, limita a utilização em números de Mersenne para $m < 257$, mesmo para os computadores PC modernos de mercado atual (clock de 1GHz com 256MB de RAM).

O programa *Mersenne v.2,* ou contrário, representa uma possibilidade real de se encontrar um número primo de Mersenne desconhecido, utilizando-se um coputador PC atual.

Essa possibilidade pode aumentar, se ao programa *Mersenne v.2* forem incorporados outros algoritmos poderosos, como por exemplo a Transformada Rápida de Fourier (FFT), para se multiplicar grandes inteiros rapidamente e outros métodos estatísticos que agilizem a busca. Essas e outras possíveis melhorias ficam como sugestão para um trabalho futuro.

7 BIBLIOGRAFIA

1. Arndt, J. **"FFT Code and Related Stuff."** http://www.jjj.de/fxt/.
2. Bell Laboratories**. "Netlib FFTPack."** http://netlib.bell-labs.com/netlib/fftpack/.
3. Bracewell, R. *The Fourier Transform and Its Applications, 3rd ed.* New York: McGraw-Hill, 1999.
4. Blahut, R. E. *Fast Algorithms for Digital Signal Processing.* New York: Addison-Wesley, 1984.
5. Conway, J. H.; Guy, R. K. "Mersenne's Numbers." In *The Book of Numbers.* New York: Springer-Verlag, pp. 135-137, 1996.
6. Conway, J. H.; Guy, R. K. *The Book of Numbers.* New York: Springer-Verlag, pp. 127-130, 1996.
7. Crandall, R.; Pomerance, C. *Prime Numbers.* New York: Springer-Verlag, 2001.
8. Devlin, K. "World's Largest Prime." **FOCUS: Newsletter Math. Assoc. Amer. 17**, 1, Dec. 1997.
9. Ellison, W. J.; Ellison, F. *Prime Numbers.* New York: Wiley, 1985.
10. Flannery, S.; Flannery, D. *In Code: A Mathematical Journey.* London: Profile Books, pp. 38-42, 2000
11. Sloane, N. J. A. Sequences A003010/M3494 in **"The On-Line Encyclopedia of Integer Sequences."** http://www.research.att.com/~njas/sequences/.