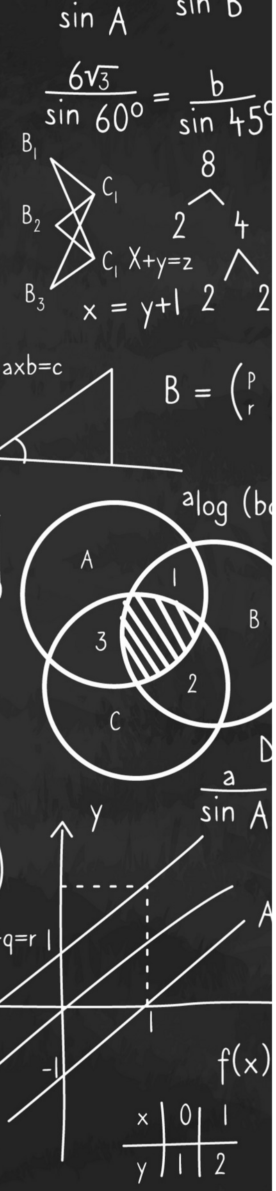


MATEMÁTICA DISCRETA 2

Aula 16

Teorema de Wilson
Algoritmo de Pollard Rho

Cristiane Loesch

Brasília
2025

Teorema de Wilson

Se p é primo então $(p-1)! \equiv -1 \pmod{p}$

EXEMPLO:

Example. Let $p=7$. We have $(7-1)! = 6! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6$. We will rearrange the factors in the product, grouping together pairs of inverses modulo 7. We note that $2 \cdot 4 \equiv 1 \pmod{7}$ and $3 \cdot 5 \equiv 1 \pmod{7}$. Hence, $6! \equiv 1 \cdot (2 \cdot 4) \cdot (3 \cdot 5) \cdot 6 \equiv 1 \cdot 6 \equiv -1 \pmod{7}$. Thus, we have verified a special case of Wilson's theorem.

FONTE: Rosen, K. H. (1984)

Teorema de Wilson

Se p é primo então $(p-1)! \equiv -1 \pmod{p}$

Demonstração:

Proof. When $p=2$, we have $(p-1)! \equiv 1 \equiv -1 \pmod{2}$. Hence, the theorem is true for $p=2$. Now, let p be a prime greater than 2. Using Theorem 3.7, for each integer a with $1 \leqslant a \leqslant p-1$, there is an inverse $\bar{a}$, $1 \leqslant \bar{a} \leqslant p-1$, with $a\bar{a} \equiv 1 \pmod{p}$. From Proposition 3.4, the only positive integers less than p that are their own inverses are 1 and $p-1$. Therefore, we can group the integers from 2 to $p-2$ into $(p-3)/2$ pairs of integers, with the product of each pair congruent to 1 modulo p. Hence, we have

$$2 \cdot 3 \cdots (p-3) \cdot (p-2) \equiv 1 \pmod{p}.$$

FONTE: Rosen, K. H. (1984)

Teorema de Wilson

Se p é primo então $(p-1)! \equiv -1 \pmod{p}$

Demonstração (continuação):

We conclude the proof by multiplying both sides of the above congruence by 1 and $p-1$ to obtain

$$(p-1)! = 1 \cdot 2 \cdot 3 \cdots (p-3)(p-2)(p-1) \equiv 1 \cdot (p-1) \equiv -1 \pmod{p}. \quad \square$$

An interesting observation is that the converse of Wilson's theorem is also true, as the following theorem shows.

FONTE: Rosen, K. H. (1984)

Teorema de Wilson

Corolário do Teorema:

Theorem 5.1. If n is a positive integer such that $(n-1)! \equiv -1 \pmod{n}$, then n is prime.

Proof. Assume that n is a composite integer and that $(n-1)! \equiv -1 \pmod{n}$. Since n is composite, we have $n=ab$, where $1 < a < n$ and $1 < b < n$. Since $a < n$, we know that $a \mid (n-1)!$, because a is one of the $n-1$ numbers multiplied together to form $(n-1)!$. Since $(n-1)! \equiv -1 \pmod{n}$, it follows that $n \mid [(n-1)! + 1]$. This means, by the use of Proposition 1.3, that a also divides $(n-1)! + 1$. From Proposition 1.4, since $a \mid (n-1)!$ and $a \mid [(n-1)! + 1]$, we conclude that $a \mid [(n-1)! + 1] - (n-1)! = 1$. This is an obvious contradiction, since $a > 1$. □

We illustrate the use of this result with an example.

FONTE: Rosen, K. H. (1984)

Teorema de Wilson

DESAFIO: Determinar o resto não negativo de 70! mod 5183

Solução: Note que $5183 = 71 \cdot 73$. Começaremos encontrando o resíduo de 70! mod 71 e 73. Pelo Teorema de Wilson,

$$70! = -1 \pmod{71}.$$

Agora, faça $k \equiv 70! \pmod{73}$. Então,

$$\begin{aligned} 71 \cdot 72 \cdot k &\equiv 70! \cdot 71 \cdot 72 \pmod{73}, \\ (-2)(-1)k &\equiv 72! \pmod{73}, \\ 2k &\equiv -1 \pmod{73}. \end{aligned}$$

Note que $2 \cdot 37 = 74 \equiv 1 \pmod{73}$. Assim,

$$\begin{aligned} 37 \cdot 2k &\equiv 37 \cdot (-1) \pmod{73}, \\ k &\equiv -37 \pmod{73}, \\ k &\equiv 36 \pmod{73}. \end{aligned}$$

Logo, $70! \equiv -1 \pmod{71}$ e $70! \equiv 36 \pmod{73}$. Vamos agora utilizar essas duas informações para construir a congruência módulo 5183. Primeiramente, $70! \equiv -1 \pmod{71}$ significa $70! \equiv -1 + 71a$ para algum $a \in \mathbb{Z}$. Colocando isto na segunda congruência, temos

Resolução - FONTE: Rispoli. V. (2024)

Teorema de Wilson

DESAFIO: Determinar o resto não negativo de 70! mod 5183

$$\begin{aligned}
-1 + 71a &\equiv 36 \pmod{73}, \\
71a &\equiv 37 \pmod{73}, \\
-2a &\equiv 37 \pmod{73}, \\
(-37)(-2)a &\equiv (-37)(37) \pmod{73}, \\
a &\equiv -1369 \pmod{73}, \\
a &\equiv 18 \pmod{73}.
\end{aligned}$$

Ou seja, a última congruência significa que $a = 18+73b$ para algum $b \in \mathbb{Z}$. Jogando isso em $70! \equiv -1+71a$, temos

$$\begin{aligned}
70! &= -1 + 71(18 + 73b) \\
&= 1277 + 5183b,
\end{aligned}$$

portanto, $70! \equiv 1277 \pmod{5183}$.

Resolução - FONTE: Rispoli. V. (2024)

Métodos de Fatoração

- → Método de Fatoração por tentativas
- → Método De Fermat
- → Métodos de Pollard
  - → Método Rho
  - → Método p-1

Fatoração de Pollard Rho

→ método de fatoração de números inteiros

- **O problema:** Dado um número inteiro n (grande, composto) e seu menor divisor primo p, Encontrar um fator não trivial de n

→ 1975 – John M. Pollard

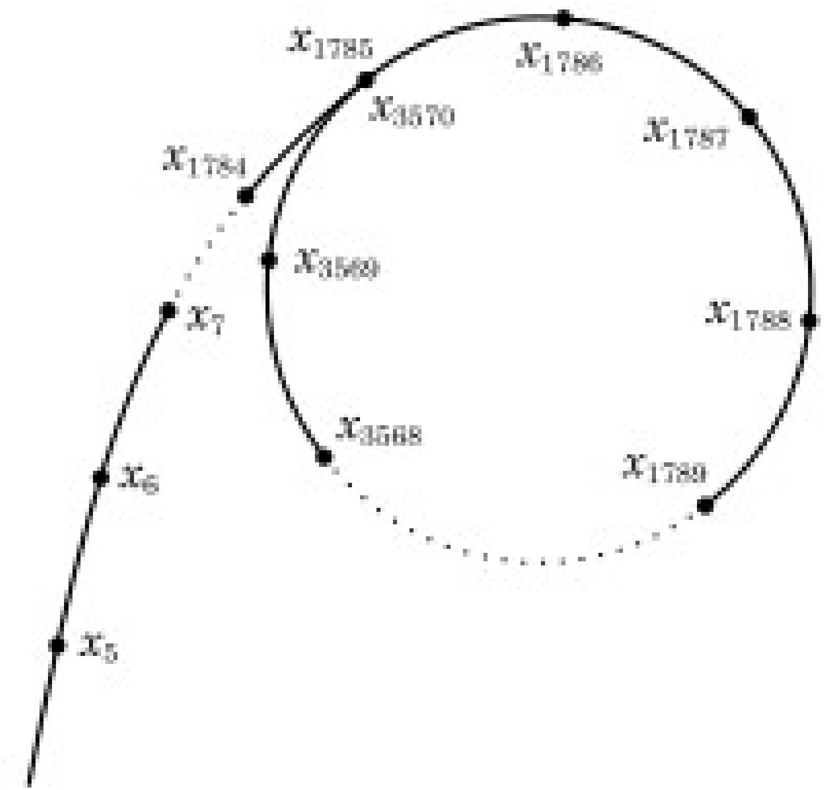
- criou algoritmos (probabilisticos) para calcular os fatores de números inteiros grandes através de uma sequência de operações polinomiais (adição, subtração e multiplicação)
    - denominados, atualmente, rho e p-1

→ Método Rho

- Inicialmente, chamado método Monte Carlo por sua natureza pseudo-aleatória, encontra um fator em menos tempo que os modelos determinísticos
- O tempo de processamento é proporcional à raiz quadrada do menor fator primo do número composto a ser fatorizado

Fatoração de Pollard Rho

→ Método Rho



FONTE: https://en.wikipedia.org/wiki/Pollard%27s_rho_algorithm

Fatoração de Pollard Rho

Método Rho

- Caso de sucesso:
  - 1980 – Número de Fermat f_8
  - Levou apenas cerca de 2 horas para realizar o cálculo, pois p é muito menor do que o outro número

p = 1238926361552897

$f_8 = 1238926361552897 \times 93461639715357977769163558199606896584051237541638188580280321.$

Onde usar?

Exemplo:

→ resolução de logarítmos discretos que requerem segurança
→ métodos de criptografia de chave pública

Fatoração de Pollard Rho

Descrição do Método:

Supõe-se que n é um inteiro grande, composto, e que p é seu menor divisor primo. O objetivo é escolher inteiros $x_0, x_1, \ldots, x_s$ de forma que estes inteiros tenham resíduos não negativos mínimos distintos, módulo n, mas seus resíduos não negativos mínimos módulo p não sejam todos distintos. Como se pode ver, usando argumentos probabilísticos (ver [9] ou [14], por exemplo), é provável que este seja o caso quando s é grande comparado a $\sqrt{p}$ mas pequeno quando comparado a $\sqrt{n}$, e os números são escolhidos randomicamente.

Uma vez que tenham sido encontrados inteiros x_i e x_j onde $0 \leq i < j \leq s$ tais que $x_i \equiv x_j \pmod{p}$ mas $x_i \not\equiv x_j \pmod{n}$, segue que $mdc(x_i - x_j, n)$ é um divisor não trivial de n, já que $x_i - x_j$ é divisível por p, mas não por n. O número $mdc(x_i - x_j, n)$ pode ser encontrado rapidamente usando-se o algoritmo de Euclides. Entretanto, encontrar $mdc(x_i - x_j, n)$ para cada par (i, j)

FONTE: Antunes, C. M. (2022) - UFRGS

Fatoração de Pollard Rho

Descrição do Método:

com $0 \leq i < j \leq s$ requer que sejam encontrados $O(s^2)$ máximos divisores comuns ([3]). A seguir, mostra-se, inicialmente, como calcular os x_i, e logo depois, como reduzir o número de vezes em que o algoritmo de Euclides precisa usado.

Para encontrar tais inteiros x_i e x_j, começa-se com um valor inicial x_0, que é escolhido randomicamente, e uma função polinomial $f(x)$ arbitrária com coeficientes inteiros e grau maior que 1. Calculam-se os termos x_k, $k = 1, 2, 3, \ldots$, usando a definição recursiva

$$x_{k+1} \equiv f(x_k) \pmod{n}, \; 0 \leq x_{k+1} < n.$$

O polinômio $f(x)$ deve ter a propriedade que a seqüência $x_0, x_1, \ldots, x_k, \ldots$ se comporta como uma seqüência verdadeiramente aleatória.[1] O exemplo a seguir ilustra como esta seqüência é gerada.

FONTE: Antunes, C. M. (2022) - UFRGS

Fatoração de Pollard Rho

EXAMPLE 4.26 Let $n = 7943$, $x_0 = 2$, and $f(x) = x^2 + 1$. Then

$$x_1 = 5, \quad x_2 = 26, \quad x_3 = 677, \quad x_4 = 5579, \quad x_5 = 4568, \quad x_6 = 364,$$
$$x_7 = 5409, \quad \dots$$

Choosing $x_0 = 2$ and $f(x) = x^2 + 1$, we generate the sequence $\{x_k\}$:

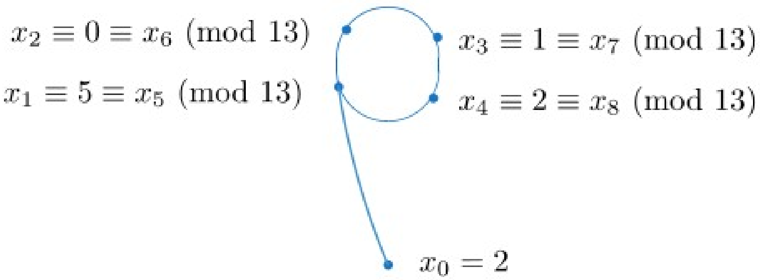
$$2, 5, 26, 677, 5579, 4568, 364, 5409, \dots$$

when reduced modulo 13, yields the periodic sequence

$$2, \underbrace{5, 0, 1, 2}, \underbrace{5, 0, 1, 2}, \underbrace{5, 0, 1, 2}, 5, 0, \dots$$

with period 4.

This periodic behavior can be displayed pictorially, as in Figure 4.2. Since it resembles the Greek letter ρ (rho), the factoring method is now known as the **rho method**.



FONTE – Adaptada de Koshy, T. (2007)

Fatoração de Pollard Rho

Método Rho – Periodicidade

O algoritmo de Rho utiliza-se de uma função interativa, f(x), gerando uma sequência de valores $x_{i.}$

Observe que tal função é limitada no conjunto Z_n (um conjunto finito) , assim tais valores x_i, eventualmente, devem começar a se repetir, tal repetição/periodicidade do algoritmo é importante para o processo, pois permite o teste do MDC.

Além disso, como falamos de um algoritmo probabilístico, tal característica pode auxiliar na “previsão” de certos padrões no processo de fatoração.

Fatoração de Pollard Rho

Resumindo:

1) N

2) Definir um polinômio diofantino (de grau maior ou igual a 2),

$$f(x) = x^2+a, \; a \neq 0 \text{ (por exemplo)}$$

3) Escolher uma semente x_0

4) Gerar uma sequência pseudo-aleatória de números, a partir de x_0 e f(x)

$$x_{k+1}=f(x_k) \bmod n$$

5) Encontrar o fator D de N garantindo que D < < N , D≠1, D|N

$$D = mdc(|x_j-x_i| , N)$$

pois $x_i \equiv x_j \bmod D$, $x_i \neq x_j$, $i < j$, logo $D|\, x_j-x_i$

Fatoração de Pollard Rho

A Refined Version

Since $x_i \equiv x_j \pmod{d}$,

$$x_{i+1} \equiv f(x_i) \equiv f(x_j) \equiv x_{j+1} \pmod{d}$$

where $i < j$. Consequently, the elements of the sequence $\{x_k\}$ reduced modulo d repeat in every block of $j - i$ elements; that is, $x_r \equiv x_s \pmod{d}$, where $r \equiv s \pmod{j-i}$, and $r, s \geq i$. In fact, $\{x_k\}$ reduced modulo d is periodic with period that is a factor of $j - i$.

In particular, let t be the smallest multiple of $j - i$ that is greater than i. Then $t \equiv 0 \pmod{j-i}$; so $2t \equiv t \pmod{j-i}$. Consequently, $x_t \equiv x_{2t} \pmod{d}$. Thus, to find a nontrivial factor of n, we compute the gcd's $(x_{2k} - x_k, n)$, where $k \geq 1$, as the next example demonstrates.

$D = mdc(|x_{2i} - x_i| , N)$

FONTE: Koshy, T. (2007)