ANTECESSORES RELATIVAMENTE PRIMOS

ALUNO: MÁRCIO PALMARES

Trabalho apresentado para a Disciplina CM141 - TÓPICOS DE MATEMÁTICA II, a cargo do Prof. ABEL SOARES SIQUEIRA

INTRODUÇÃO

Neste trabalho, *número natural* significa *inteiro positivo*, isto é, $\mathbb{N} = \{1, 2, 3, ...\}$.

Dado um natural n, em certos problemas de Teoria de Números é necessário considerar o conjunto formado pelos antecessores k de n tais que k e n não possuem divisores comuns, ou, equivalentemente, tais que $\text{mdc}(k, n) = 1$. Tais pares de números são chamados *primos entre si*, *coprimos* ou *relativamente primos*. Adotamos a sigla ARP como abreviatura para a expressão *antecessores relativamente primos*. Em símbolos,

$$\text{ARP}(n) = \{\, k \in \mathbb{N} \mid 1 \leq k < n \text{ e } mdc(k, n) = 1 \,\}.$$

Por exemplo, para $n = 10$, o conjunto de seus ARP é:

$$\text{ARP}(10) = \{\, 1, 3, 7, 9 \,\}.$$

Observemos que em ARP(10) nem todos os elementos são primos: 1 não é primo, por definição, e 9 é composto, múltiplo de 3; no entanto, mdc(1, 10) = mdc(9, 10) = 1.

Para $n = 30$,

$$\text{ARP}(30) = \{\, 1, 7, 11, 13, 17, 19, 23, 29 \,\}.$$

Neste caso, como se vê, com exceção do 1, os demais elementos em ARP(30) são primos. Mais ainda, observamos que podemos organizar tais elementos em pares ordenados cuja soma das componentes é $n = 30$:

(1, 29)
(7, 23)
(11, 19)
(13, 17)

Tais pares correspondem à ação de "somar os extremos" da sequência 1, 7, 11, 13, 17, 19, 23, 29, o mesmo procedimento que adotamos inicialmente quando queremos calcular a soma dos termos de uma progressão aritmética, como 2, 4, 6, 8. Em ambos os casos a soma dos extremos é constante. Notemos ainda que mdc(1, 29) = mdc(7, 23) = mdc(11, 19) = mdc(13, 17) = 1.

Finalmente, observamos que na sequência 1, 7, 11, 13, 17, 19, 23, 29 a diferença entre dois termos sucessivos (primeira derivada discreta) fornece-nos um "padrão de crescimento"

$$6,\ 4,\ 2,\ \mathbf{2},\ 2,\ 4,\ 6$$

que apresenta simetria e cuja derivada discreta (segunda derivada discreta em termos da sucessão inicial), a partir do centro, para a esquerda ou para a direita, é em módulo constante e igual a 2. Um **problema pendente** —ainda não examinado pelo autor desse trabalho— é o de **saber se um polinômio interpolador** (que pode ser obtido por integração discreta) **poderia nos fornecer algum tipo de informação sobre a distribuição dos primos no conjunto dos ARP de qualquer múltiplo de 30**. Essa hipótese deve-se a que o mesmo padrão de crescimento é preservado quando consideramos múltiplos de 30. Por exemplo, para obter ARP(60) é suficiente adicionarmos mais uma "coluna" à sucessão original, formando um conjunto de pares ordenados (ou um conjunto comum ou uma matriz 8x2) e preservando as relações $\text{mdc}(a, b) = 1$ e $a + b = n$:

(1, 59)
(7, 53)
(11, 49)
(13, 47)
(17, 43)
(19, 41)
(23, 37)
(29, 31)

Observemos que:

(1) O único número composto no "conjunto" acima é 49;
(2) A segunda coluna (a segunda componente dos pares) segue o mesmo padrão de crescimento da primeira coluna (porém com sentido contrário);
(3) O centro de simetria do padrão de crescimento da sucessão completa continua sendo igual a 2.

Esse comportamento altamente organizado ou previsível dos ARP de 30 e de seus múltiplos sugere-nos o desenvolvimento de métodos computacionais para produzir grandes coleções de ARP e pesquisar o comportamento dos primos em seu interior. Sabe-se (até o momento) que a distribuição dos primos é aleatória. No entanto, introduzindo alguns "poucos" números compostos em sucessões específicas, conseguimos distribuições mistas (de compostos e primos) que respeitam padrões determinados, como vemos no caso dos ARP(60). Uma questão a ser respondida é então: listas "puras" de primos (obtidas de conjuntos de ARP após a eliminação de compostos) poderiam exibir algum padrão? Se a resposta for afirmativa, poderíamos afirmar que os primos se distribuem de acordo com padrões em subconjuntos específicos de $\mathbb{N}$.

1. A função ϕ de Euler: o número de elementos em ARP(n)

O número de elementos em ARP(n) é calculado pela ***função ϕ de Euler***, cuja expressão é:

$$\phi(n) = n * \prod_{p|n} (1 - \frac{1}{p}) = n * \prod_{p|n} (\frac{p-1}{p}).$$

onde p é um fator primo da decomposição de n.

Por exemplo, para $n = 30$, temos:

$$\phi(30) = 2 * 3 * 5 \prod_{p|n} (\frac{p-1}{p}) = 2 * 3 * 5 * (\frac{1}{2}) * (\frac{2}{3}) * (\frac{4}{5}) = 8.$$

O número de ARP de um natural n aparece em diferentes contextos, por exemplo, como o grau do n-ésimo polinômio ciclotômico (associado às raízes primitivas de ordem n da unidade, em $\mathbb{C}$) e também na equação abaixo, que nos fornece um interessante modo de particionar um natural n:

$$n = \sum_{d|n} \phi(d),$$

onde d é um divisor de n.

Para $n = 10$, temos

$$10 = \sum_{d|10} \phi(d) = \phi(1) + \phi(2) + \phi(5) + \phi(10) = 1 + 1 + 4 + 4.$$

Para construir essa função usando a linguagem Julia, copiamos parte do raciocínio usado por outros programadores, substituindo, entretanto, o emprego do Crivo de Eratóstenes pelo emprego da função *factor*, disponível no pacote *Primes*, que retorna os fatores primos da decomposição de n no "formato" desejado (num vetor/matriz, num "dicionário" ou num conjunto). Eis o código (é preciso instalar e carregar o pacote *Primes*):

```
function phi(n::Integer)

    m = n
    for p in factor(Set, n)
        m = m - div(m,p) # m = m * (1 - 1/p)
    end
    return Int(round(m))
end
```

2. A função ARP(n)

Para construir o conjunto dos ARP de um n qualquer, criamos antes a função ***coprimos***, que diz se dois números dados são relativamente primos ou não:

```
function coprimos(n, m::Integer)

    if gcd(n,m) == 1
        return true
    else
        return false
    end
end
```

Neste caso usamos a função disponível $\gcd(a, b)$, máximo divisor comum dos inteiros a e b.

A função que nos fornece o conjunto dos ARP de n é então obtida da seguinte maneira:

```
function ARP(n::Integer)

A = Vector()

    for k = 1:n-1
        if coprimos(k,n)
            A = push!(A, k)
        end
    end
    return A
end
```

Observemos que a função retorna um vetor (matriz coluna) e não um conjunto propriamente dito.

Para facilitar a visualização de conjuntos pequenos de ARP, criamos ainda a função

```
function ARP_ver(n::Integer)

    for p = 1:Int(n/2)
        if p in ARP(n)
            println("($(p), $(n-p))")
        end
    end
end
```

que imprime a lista de ARP organizados em pares do tipo (a, b) onde $\text{mdc}(a, b) = 1$ e $a + b = n$.

3. Eliminando os números compostos e o 1 de ARP(n)

Ao eliminar os compostos e o número 1 de ARP(n), obtemos uma sucessão pura de primos. Acrescentando a ela os primos que estão na decomposição de n, obtemos

a lista de ***primos até n***, dada pela função

```
function primosate(n::Integer)

    X = ARP(n)
    Y = filter(isprime, ARP(n))
    S = unique(factor(Vector, n))
    T = flipdim(S,1)

    for p in T

        Y = unshift!(Y, p)
    end
    return Y
end
```

Criamos também a função ***primos até n–visualização***, que produz uma matriz mais adequada para a visualização (sempre que o número de primos até n não for primo!):

```
function primosate_vis(n::Integer)

    A = ARP(n)
    B = primosate(n)
    C = length(B)
    D = unique(factor(Vector, C))
    k = maximum(D)

    reshape(B, k, Int(C/k))
end
```

Rodando essa última função num computador comum, conseguimos resposta imediata para $n = 10^6$ (dez milhões). No entanto, para $n = 10^7$ (cem milhões) o programa falha, talvez por falta de memória para armazenar os primos.

4. Conclusão

Num programa como o Excel, por exemplo, é possível construir matrizes de ARP facilmente, pois novas colunas podem ser obtidas das anteriores por simples adição. Por exemplo, ainda no caso do 30 e de seus múltiplos, a primeira coluna da matriz seria formada pelos elementos em ARP(30), a segunda coluna seria obtida por subtração: $[a]_{i2} = 60–[a]_{i1}$. Para obter as próximas colunas basta somar 60 à penúltima coluna. **Não conseguimos descobrir ainda como fazer isso na linguagem Julia**. No entanto, no momento em que conseguirmos, o próximo passo seria construir um meio de determinar, pela coordenadas dos elementos da matriz, as posições de todos os múltiplos de 7, por exemplo. Depois, as posições de todos os múltiplos de 11, e assim sucessivamente. Isso é possível porque os múltiplos de 7 dentro do conjunto de ARP dos múltiplos de 30 se distribuem de acordo com um padrão determinado (o mesmo que gera a primeira sucessão). Essa distribuição,

quando observada graficamente (como números coloridos numa matriz), apresenta simetrias (a partir da sétima coluna da matriz, para o 7; a partir da décima primeira coluna para o 11, etc.) e os múltiplos em questão são obtidos por reflexão do padrão em torno do eixo de simetria. Assim, quando chegarmos à coluna cujo número é 215.656.441 (o mínimo múltiplo comum de 1, 7, 11, 13, 17, 19, 23, 29), bastará refletir o padrão conjunto produzido pelos compostos múltiplos desses primos, e então obteremos "gratuitamente" compostos do outro lado do eixo de simetria. Após eliminá-los, precisaremos considerar compostos múltiplos da segunda coluna, por um processo similar. Repetindo esse raciocínio, sobrarão apenas primos do outro lado do centro de simetria, até um certo n, obtidos sem testes usuais de primalidade. Essa ideia precisa ainda ser implementada, mas para isso é preciso ter mais conhecimento de programação, algo que pretendemos adquirir a partir de agora.

References

1. Rotman, Joseph J., *A first course in abstract algebra : with aplications*, 3rd ed., Pearson Prentice Hall, 2006.
2. Milies, Francisco César Polcino, *Números: Uma introdução à Matemática*, 3rd ed., São Paulo: Editora da Universidade de São Paulo, 2003.
3. Birkhoff, Garrett, *Álgebra Moderna Básica*, 4rd ed., Rio de Janeiro: Editora Guanabara Dois, 1980.