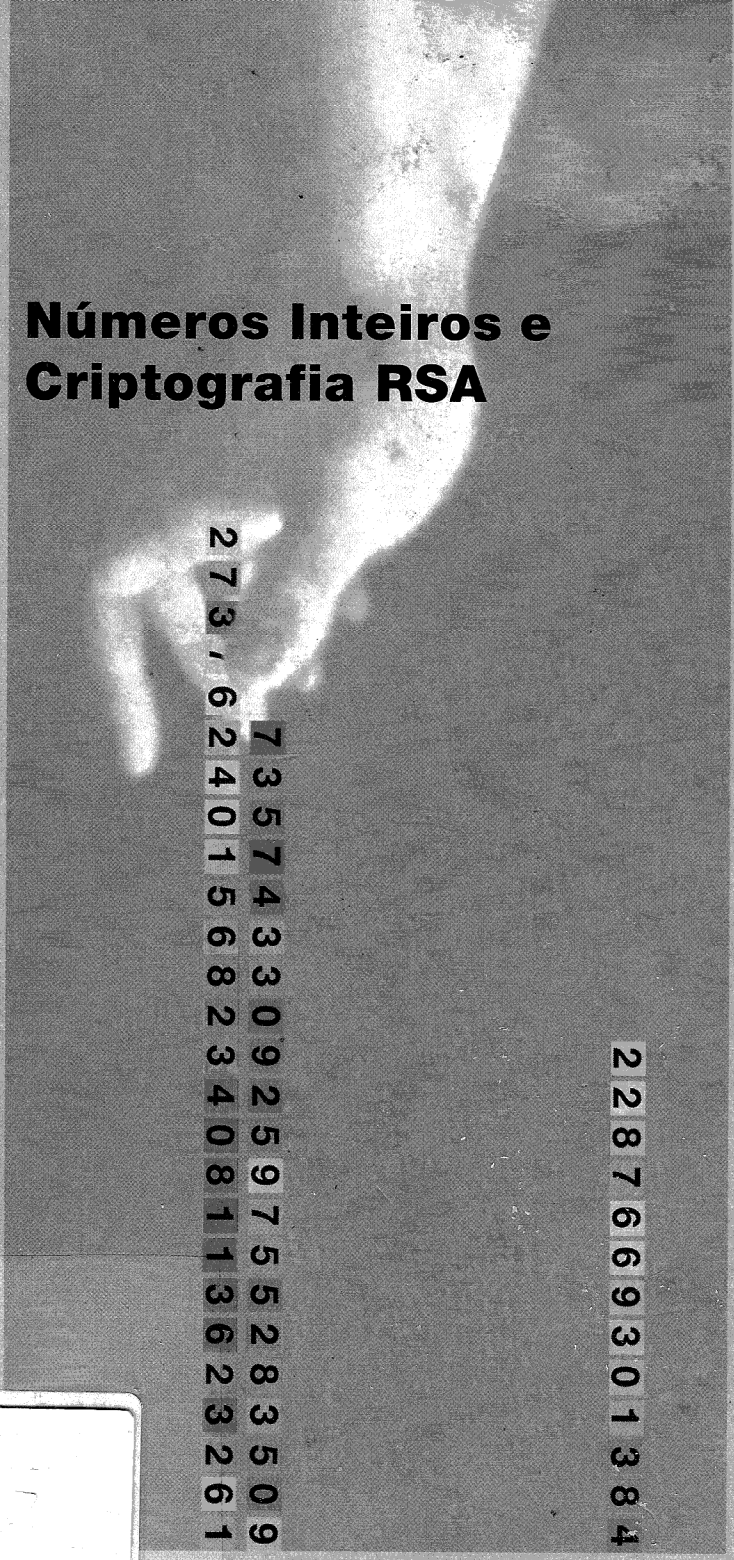
S. C. Coutinho

Números Inteiros e Criptografia RSA



Instituto Nacional de Matemática Pura e Aplicada - IMPA
Série de Computação e Matemática

S. C. Coutinho

Números Inteiros e Criptografia RSA

Segunda Edição
(Terceira impressão)

Instituto Nacional de Matemática Pura e Aplicada - IMPA
Série de Computação e Matemática

Números Inteiros e Criptografia RSA
Copyright ©, 2005 by **S. C. Coutinho**

Direitos reservados, 2005 pela Associação
Instituto Nacional de Matemática Pura e Aplicada, IMPA
Estrada Dona Castorina, 110
22460-320, Rio de Janeiro, RJ

Impresso no Brasil/Printed in Brazil
Segunda edição
Terceira impressão

Capa: Rodolfo Capeto e Noni Geiger

Coutinho , S. C.
Números Inteiros e Criptografia RSA
Rio de Janeiro, IMPA, 2005
226 páginas (Série de Computação e Matemática)
Bibliografia
1. Teoria dos números. I. Série. II. Título.
CDD-512.7
ISBN: 85-244-0124-9

Série de Computação e Matemática:
Comissão Editorial:
Paulo Cezar Carvalho
Marcelo Gattass
Jonas Gomes (coordenador)
Luiz Velho

Títulos publicados:

- Computação Gráfica: Imagem - *Jonas Gomes, Luiz Velho*
- Números Inteiros e Criptografia RSA - *S.C. Coutinho*
- Computação Gráfica, vol. 1 - *Jonas Gomes e Luiz Velho*
- Sistemas Gráficos 3D - *Luiz Velho e Jonas Gomes*
- Códigos Corretores de Erros - *Abramo Hefez e Maria Lucia T. Villela*
- Fundamentos da Computação Gráfica - *Jonas Gomes e Luiz Velho*

Distribuição:
IMPA
Estrada Dona Castorina, 110
22460-320 Rio de Janeiro, RJ
e-mail: ddic@impa.br
http://www.impa.br

Para Andrea e Daniel,
Meu outro mundo

Conteúdo

Prefácio

Introdução 1
1. Criptografia 1
2. Criptografia RSA 3
3. Computação algébrica 5
4. Teoria dos números 8
5. Fermat, Euler e Gauss 9
6. O livro 13
7. Teoremas e demonstrações 14
8. Pré-requisitos 16

Capítulo 1 Algoritmos fundamentais 17
1. Algoritmos 17
2. Algoritmo de divisão 19
3. Teorema de divisão 22
4. Algoritmo euclidiano 23
5. Demonstração do algoritmo euclidiano 26
6. Algoritmo euclidiano estendido 28
7. Exercícios 32

Capítulo 2 Fatoração única 35
1. Teorema da fatoração única 35
2. Existência da fatoração 36
3. Eficiência do algoritmo usual de fatoração 39
4. Fatoração por Fermat 40
5. Demonstração do algoritmo de Fermat 41
6. Propriedade fundamental dos primos 43
7. Números irracionais 44
8. Unicidade 46
9. Exercícios 48

Capítulo 3 Números primos 53
1. Fórmulas polinomiais 53
2. Fórmulas exponenciais: números de Mersenne 56
3. Fórmulas exponenciais: números de Fermat 57
4. Fórmulas fatoriais 59
5. Infinidade dos primos 60
6. Crivo de Eratóstenes 62
7. Exercícios 65

Capítulo 4 Aritmética modular 69
1. Relações de equivalência 69
2. Inteiros módulo n 72
3. Aritmética modular 74
4. Critérios de divisibilidade 77
5. Potências 79
6. Equações diofantinas 80
7. Divisão modular 82
8. Exercícios 85

Capítulo 5 Indução e Fermat 87
1. Hanói! Hanói! 87
2. Indução finita 91
3. Pequeno teorema de Fermat 93
4. Contando raízes 96
5. Exercícios 100

Capítulo 6 Pseudoprimos 105
1. Pseudoprimos 105
2. Números de Carmichael 107
3. Teste de Miller 111
4. Primalidade e computação algébrica 113
5. Exercícios 115

Capítulo 7 Sistemas de congruências 117
1. Equações lineares 117
2. Um exemplo astronômico 118
3. Algoritmo chinês do resto 120
4. Módulos não co-primos 123
5. Potências, novamente 125
6. Partilha de senhas 126
7. Exercícios 129

Capítulo 8 Grupos 133
1. Definição e exemplos 133
2. Simetrias 137
3. Interlúdio 138
4. Grupos aritméticos 141
5. Subgrupos 144
6. Subgrupos cíclicos 145
7. Determinando subgrupos 147
8. Teorema de Lagrange 148
9. Exercícios 150

Capítulo 9 Mersenne e Fermat 155
1. Números de Mersenne 155
2. Números de Fermat 158
3. Fermat, novamente 160
4. O teste de Lucas-Lehmer 161
5. Exercícios 165

Capítulo 10 Raízes primitivas 167
1. Teste de Lucas 167
2. Outro teste determinístico de primalidade 170
3. Números de Carmichael 172
4. Preliminares 173
5. Teorema da raiz primitiva 175
6. Calculando ordens 176
7. Exercícios 178

Capítulo 11 Criptografia RSA 181
1. Pré-codificação 181
2. Codificando e decodificando 182
3. Por que funciona? 184
4. Por que o RSA é seguro? 186
5. Escolhendo primos 187
6. Assinaturas 189
7. Exercícios 191

Epílogo 195
1. Criptografia e teoria dos números 195
2. Teoria de grupos 196
3. Computação algébrica 197

Apêndice Algoritmos Complementares 199

1. Raízes quadradas 199
2. Potenciação módulo n 201

Bibliografia 203
Índice dos principais algoritmos 206
Índice dos principais resultados 207
Índice 208

Prefácio

Este livro tem uma meta muito precisa: descrever em detalhes um dos métodos de criptografia mais usados em aplicações comerciais, o RSA. Para compreendê-lo é preciso ter bastante familiaridade com as noções de máximo divisor comum, números primos, fatoração, aritmética modular e função totiente, entre outras. O objetivo deste livro é, portanto, o de fornecer a base matemática necessária ao entendimento e aplicação do método RSA. A maior parte dos conceitos e métodos de que vamos precisar faz parte da área da matemática conhecida como teoria de números.

A descrição acima é fiel ao espírito deste livro, mas revela apenas uma de suas facetas; o que poderíamos chamar de ponto de vista do estudante de informática. É fácil dar uma outra descrição, igualmente fiel, e melhor adaptada ao estudante de matemática. Deste outro ponto de vista, este livro é uma introdução elementar à teoria de números e às estruturas algébricas, com ênfase em resultados construtivos e métodos algorítmicos. Assim procuramos estudar conceitos abstratos através de suas manifestações concretas, e as demonstrações que conduzem a algoritmos efetivos foram preferidas às puramente existenciais.

Sem dúvida que, para muitos, este livro vai servir principalmente como um guia à criptografia matemática. Mas é importante não esquecer que quase toda a matemática utilizada no RSA foi desenvolvida sem nenhuma finalidade prática. Para Fermat, Euler, Gauss ou Riemann, o estudo das propriedades dos números inteiros era apenas uma área fascinante da matemática. Nela combinam-se resultados muito simples com demonstrações de enorme engenhosidade. Isto nos deve ajudar a lembrar que a matemática tem vida própria, e que suas aplicações, por muitas que sejam, são apenas uma faceta de seu enorme impacto na história cultural da humanidade.

Neste espírito, o livro não se restringe à descrição mais econômica possível do RSA, mas procura inserir as técnicas matemáticas requeridas em um esquema mais amplo. Por exemplo, tratamos os testes de primalidade como uma aplicação da *teoria de grupos*. Com um pouco de esforço poderíamos fazer desaparecer os grupos, reduzindo assim as demonstrações a cálculos com fórmulas mais ou menos elementares. Entretanto, um dos efeitos desta 'simplificação' seria uma apresentação mais difícil de motivar; que é uma desvantagem. Além disso, o uso dos grupos trás à tona as semelhanças en-

tre os testes de primalidade e alguns métodos de fatoração (de números de *Mersenne* e *Fermat*), o que é uma grande vantagem. Isto sem mencionar as inúmeras aplicações dos grupos a outras áreas.

Este livro foi desenvolvido a partir de notas de aulas escritas para o curso de *álgebra para a informática* da Universidade Federal do Rio de Janeiro. Seu público original foram os estudantes de *primeiro período* de informática e matemática. Isto determinou muitas das características do livro, e é responsável por algumas de suas idiossincrasias. Em primeiro lugar, os requisitos assumidos são mínimos; e são discutidos em detalhe na seção 8 da introdução. O texto também não pressupõe nenhum conhecimento prévio de computação. Entretanto, uma certa experiência com noções básicas de programação dará melhor perspectiva a muitas das questões tratadas aqui. Por isso alguns exercícios simples de programação são incluídos ao final de cada capítulo. A maior parte destes programas gera dados a serem comparados com alguma conjectura ou resultado da teoria de números, de modo que são o que poderíamos chamar de *'experimentos matemáticos'*.

Um comentário sobre estilo. Freqüentemente os livros de matemática são escritos como uma monótona sucessão de definições, teoremas e demonstrações. Esta é uma penosa herança d'*Os Elementos* de Euclides, que foi por muitos séculos o livro de matemática mais publicado no Ocidente. É bom não esquecer que nem mesmo entre os matemáticos da Grécia Antiga este estilo marmóreo prevaleceu. Arquimedes, por exemplo, menciona em seus tratados as dificuldades que teve, as derrotas que sofreu, e até enuncia proposições que pensou verdadeiras, e mais tarde descobriu serem falsas, como um alerta a outros matemáticos. Neste livro, preferi o exemplo de Arquimedes ao de Euclides. Por isso o estilo é bastante informal, com muitas notas históricas entremeadas ao texto. A discussão dos tópicos mais difíceis começa com um exemplo característico, e não com um mergulho em uma seqüência abstrata de axiomas. Também não hesitei em descrever os algoritmos com instruções em português, e um mínimo de símbolos; nem fiz esforço para otimizá-los. Achei mais importante ser compreendido. A experiência mostra que programar estes algoritmos não oferece maiores dificuldades, mesmo ao iniciante.

Uma outra característica do livro é a maneira como é feita referência aos teoremas, proposições e algoritmos. É costumeiro numerá-los e referir-se a eles pelo número. Entretanto, boa parte dos resultados deste livro têm nomes tradicionais, e é por esses nomes que são mencionados no correr do texto. Os poucos resultados sem nome são citados pela seção e capítulo onde podem ser encontrados. Isto dificulta um pouco a leitura de quem deseja consultar o livro apenas sobre um tópico específico. Por isso achei melhor incluir índices dos principais resultados e algoritmos, relacionados em ordem alfabética, e

indicando a página onde podem ser encontrados.

O texto do livro foi utilizado, de forma mais ou menos completa, como notas de aula para estudantes de Informática da UFRJ entre 1994 e 1997. Seções inteiras das notas foram reescritas com base em sugestões de estudantes, cuja influência no conteúdo, enfoque, e apresentação do material foi absolutamente fundamental. Por isso, é aos alunos e alunas que assistiram à criação deste livro a partir de notas avulsas de aula que vão, em primeiro lugar, os meus agradecimentos. Muitas outras pessoas leram o texto, ou parte dele, e ofereceram sugestões; gostaria de agradecer principalmente a S. Toscano de Melo, Cláudio César Saccomori Júnior, Bernardo Felzenszwalb, Amilcar Pacheco, Paulo A. Burlandy, Renata M. da Rosa e Lorenzo Diaz Casado. Um agradecimento especial ao Fernando C.M. de Faria, cujo entusiasmo por números de Fermat me levou a resgatar da literatura muitos algoritmos e resultados interessantes, que acabaram incluídos no livro. Finalmente, agradeço a Jonas Gomes pelo convite para publicar o livro na Série de Computação e Matemática e pelo empenho durante o processo de publicação.

O texto desta segunda edição foi consideravelmente melhorado graças às muitas correções e sugestões que me foram enviadas; sou especialmente grato a Keith Matthews, Ruy Carlos Miritz e Daniel Sadoc Menasché pelas suas contribuições.

Rio de Janeiro, julho de 2000

S. C. Coutinho

Introdução

O foco deste livro é o método de criptografia de chave pública conhecido como RSA. Toda a matemática que vamos estudar estará ligada, direta ou indiretamente, a este método. A introdução descreve brevemente o método e um pouco da história da principal área da matemática que lhe serve de fundamento: a teoria de números.

1. Criptografia

Em grego, *cryptos* significa secreto, oculto. A criptografia estuda os métodos para codificar uma mensagem de modo que só seu destinatário legítimo consiga interpretá-la. É a arte dos 'códigos secretos', que todos já praticamos quando criança. O mais simples destes códigos consiste em substituir uma letra pela seguinte; isto é transladar o alfabeto uma casa para diante. Um código semelhante foi usado por César para comunicar-se com as legiões em combate pela Europa. Este parece ter sido o primeiro exemplo de um código secreto de que se tem notícia.

A criptografia tem uma irmã gêmea na arte de decifrar códigos secretos, ou *criptoanálise*. Naturalmente todo código vem acompanhado de duas receitas: uma para codificar uma mensagem; outra para decodificar uma mensagem codificada. Decodificar é o que um usuário legítimo do código faz quando recebe uma mensagem codificada e deseja lê-la. Já decifrar significa ler uma mensagem codificada *sem ser um usuário legítimo*. Portanto para decifrar é preciso 'quebrar' o código. Observe que estamos adotando um significado técnico preciso que distingue decifrar de decodificar.

Códigos como o de César padecem de um grande mal, são muito fáceis de decifrar. Na verdade, qualquer código que envolva substituir cada letra sistematicamente por outro símbolo qualquer sofre do mesmo problema. Isto se deve ao fato de que a freqüência média com que cada letra é usada em uma língua é mais ou menos constante. Por exemplo, na língua portuguesa:

- as vogais são mais freqüentes que as consoantes;
- a vogal mais freqüente é o A;
- se um monossílabo tem uma única letra, então esta letra é uma vogal;

- consoantes como S e M são mais freqüentes que as outras.

Assim, apenas contando a freqüência de cada símbolo no texto, podemos descobrir a que letra correspondem os símbolos mais freqüentes. Isto geralmente é suficiente para decifrar toda a mensagem. Observe, entretanto, que este método só funciona se a mensagem for longa. É fácil escrever uma mensagem curta cuja contagem de freqüência seja totalmente diferente da contagem média do português.

O método de *contagem de freqüência* de caracteres também pode ser usado para decifrar inscrições antigas. O exemplo mais famoso é o da decifração dos hieróglifos egípcios por J.-F. Champollion em 1822. A chave para a decifração foi a descoberta da *pedra de Roseta*, um bloco de basalto negro que está atualmente no Museu Britânico, em Londres. A pedra contém uma mesma inscrição escrita em hieróglifos, demótico e grego.

Na época de Champollion muito se discutia que tipo de escrita seriam os hieróglifos: se ideográfica, silábica ou alfabética. Em uma escrita ideográfica os símbolos representam idéias; como acontece no chinês. Nas escritas silábicas, cada símbolo vale uma sílaba. As inscrições gregas mais antigas foram escritas em um silabário conhecido como *Linear B*, decifrado em 1954. Assim em uma escrita ideográfica cada palavra corresponde essencialmente a um símbolo. Imagine que traduzimos um texto escrito em grego (que é alfabético) para uma escrita ideográfica. Esperamos, então, que o número de símbolos usados na tradução corresponda aproximadamente ao número de palavras no texto grego.

Por isso, Champollion começou seu processo de decifração contando caracteres nas inscrições da pedra de Roseta. Ele descobriu que havia 486 palavras no texto grego e 1419 caracteres no texto em hieróglifos. Portanto a escrita dos antigos egípcios não podia ser ideográfica. Hoje sabemos que os hieróglifos formam um sistema misto. Há caracteres ideográficos, caracteres silábicos e os chamados determinativos. Estes últimos servem para distinguir homônimos, indicando a que classe pertencem. Por exemplo, se escrevêssemos português usando hieróglifos, poderíamos distinguir a fruta manga da manga da camisa usando o determinativo de fruta aposto à primeira.

É claro que decifrar uma mensagem por contagem de freqüência é ainda mais simples se temos um computador. Supondo que a língua é conhecida, a maior parte do processo pode ser automatizado. Isto torna essencialmente inviáveis todos os códigos que envolvem substituição de letras. Na verdade, alguns dos primeiros computadores foram montados exatamente para au-

xiliar na decifração dos códigos secretos usados pelos alemães durante a 2ª Guerra Mundial. Um dos cientistas responsáveis por este projeto foi Alan Turing, o idealizador da *máquina de Turing* e pioneiro da inteligência artificial.

Hoje em dia, a comunicação entre computadores pela *Internet* vem criando novos desafios para a criptografia. Como é relativamente fácil interceptar mensagens enviadas por linha telefônica, torna-se necessário codificá-las, sempre que contenham informações sensíveis. Isto inclui transações bancárias ou comerciais, ou até mesmo uma compra feita com cartão de crédito.

Imagine, por exemplo, que uma empresa envia uma ordem a um banco sobre uma transação envolvendo milhões de reais. Dois problemas imediatamente se põem. Em primeiro lugar, é preciso proteger a mensagem para que não possa ser lida, mesmo que seja interceptada por um concorrente, ou por um ladrão de bancos. Por outro lado o banco precisa ter completa segurança de que a mensagem recebida foi enviada por um usuário da empresa. Em outras palavras, a mensagem precisa estar *assinada*.

Assim tornou-se necessário inventar novos códigos, que fossem difíceis de decifrar, mesmo com a ajuda de um computador. Estes códigos foram criados para o uso em aplicações comerciais, e não na comunicação entre espiões. Por isso os códigos modernos são todos de *chave pública*. Esta é uma idéia introduzida em 1976 por W. Diffie e M.E. Hellman da Universidade de Stanford e, independentemente, por R.C. Merkle da Universidade da Califórnia. No código usado por César, se você sabe codificar, então também sabe decodificar. Em um código de *chave pública* saber codificar não implica saber decodificar! Isto parece impossível: se sei codificar, para decodificar basta desfazer o que fiz. Na próxima seção veremos que 'desfazer' o processo de codificação pode não ser tão simples quanto parece.

2. Criptografia RSA

O mais conhecido dos métodos de criptografia de chave pública é o RSA. Este código foi inventado em 1978 por R. L. Rivest, A. Shamir e L. Adleman, que na época trabalhavam no Massachussets Institute of Technology (M.I.T.). As letras RSA correspondem às iniciais dos inventores do código. Há vários outros códigos de chave pública, mas o RSA é, atualmente, o mais usado em aplicações comerciais. Este é o método utilizado, por exemplo, no *Netscape*, o mais popular dos *softwares* de navegação da *Internet*.

A descrição completa do funcionamento do método RSA só será possível

no último capítulo deste livro. Será necessário desenvolver muitas idéias e técnicas matemáticas antes de chegar lá. Nosso propósito nesta seção é apenas de descrever o método em linhas gerais. Entretanto diremos o suficiente para explicar porque é tão difícil decifrar o código, mesmo sabendo como efetuar a codificação. Com isto poderemos também delimitar quais os problemas matemáticos que teremos que abordar para poder chegar a implementar este método criptográfico.

Para poder implementar o RSA precisamos de dois parâmetros básicos: dois números *primos* que vamos chamar de p e q. Para codificar uma mensagem usando o RSA é suficiente conhecer o produto dos dois primos, que vamos chamar de n. Para decodificar uma mensagem precisamos conhecer os primos p e q. A *chave de codificação* do RSA é portanto constituída essencialmente pelo número $n = pq$. Cada usuário do método tem sua própria chave de codificação. Esta chave é tornada pública: todos ficam sabendo que, para mandar uma mensagem para o banco Acme, deve ser usada a chave n. Por isso n também é conhecido como a *'chave pública'*. Já a *chave de decodificação* é constituída pelos primos p e q. Cada usuário tem que manter sua chave de decodificação secreta ou a segurança do método estará comprometida.

A esta altura você deve estar pensando que tem que estar mesmo faltando alguma coisa nesta descrição do RSA. Afinal se conhecemos $n = pq$ e para decodificar precisamos descobrir p e q, *basta* fatorar o número n e o código foi decifrado. De fato, decifrar o RSA é *teoricamente* muito simples: o obstáculo é de natureza tecnológica. Usando como chaves de codificação do RSA números muito grandes (de 150 algarismos ou mais), fatorar n para achar p e q, com os métodos atuais levaria alguns milhares de anos. É disto que depende a segurança do RSA; da ineficiência dos métodos de fatoração atualmente conhecidos.

Vamos recapitular:

- para implementar o RSA precisamos escolher dois primos muito grandes p e q,
- para codificar uma mensagem usamos $n = pq$,
- para decodificar precisamos conhecer p e q,
- a segurança do método vem do fato de que é difícil fatorar n para descobrir p e q, já que são números muito grandes.

Você observou que estas afirmações parecem se contradizer, quando tomadas em conjunto? Vejamos: a segurança do RSA depende da dificuldade de fatorar n. Porém para chegar a obter n precisamos encontrar dois pri-

mos grandes p e q. Para saber que um número é primo precisamos mostrar que não tem fatores diferentes de 1 e dele próprio. Mas se p é grande, vai ser impossível fatorá-lo, porque, como já dissemos, não existem algoritmos eficientes. Assim parece que a segurança do RSA é tanta que nem é mesmo possível implementá-lo!

Felizmente a contradição não existe. Esta será, possivelmente, uma das principais surpresas deste curso: é possível verificar que um número é primo ou composto sem tentar fatorá-lo. Por exemplo, sabe-se que o número

$$2^{2^{14}}+1$$

é composto, mas nenhum de seus fatores é conhecido. Como isto é possível é uma das coisas que você vai aprender no capítulo 6.

O que dissemos até aqui é suficiente para concluir que, para entender e aplicar o RSA, precisaremos conhecer bastante bem as propriedades dos números inteiros. Os principais problemas que abordaremos são:

- Como fatorar um número inteiro de maneira eficiente?
- Como determinar se um dado inteiro é primo?

Como já observamos, o segundo problema *não* é uma conseqüência do primeiro. A parte da matemática que estuda as propriedades dos números inteiros chama-se *teoria de números*. Esta é uma das áreas mais antigas da matemática. De fato, o primeiro dos problemas acima está entre os mais antigos problemas matemáticos abordados pela humanidade.

3. Computação algébrica

Há ainda uma questão prática que precisamos considerar. Já vimos que a chave pública do RSA é criada multiplicando-se dois primos grandes, de 60 ou mais algarismos, digamos. É claro que não é possível calcular com números deste tamanho com lápis e papel; é necessário usar um computador. Mas se você tem alguma experiência de programação, sabe que uma linguagem como *Pascal* ou *C* não permite lidar diretamente com números tão grandes. A situação se agrava quando lembramos que, para determinar primalidade, será necessário efetuar divisões com números desta escala.

Uma solução é apelar para um sistema de computação algébrica. A *computação algébrica ou simbólica* trata do cálculo exato com inteiros, frações, polinômios e até mesmo algumas funções transcendentes, como senos, cosenos e exponenciais. Neste contexto, *'cálculo exato'* significa, por exemplo, que as frações não serão expressas como decimais truncadas, como fazem

quase todas as máquinas de calcular. Assim, no domínio da computação algébrica, calculamos com frações na forma numerador sobre denominador, como faríamos com lápis e papel.

Métodos para o cálculo exato com inteiros, frações e várias funções importantes são implementados em sistemas de computação algébrica, como o *Mathematica*, o *Maple*, o *Derive* e o *Axiom*. Em particular, é possível calcular num tal sistema com inteiros de centenas de algarismos. É importante que você tenha uma idéia, ainda que superficial, de como é possível representar e calcular com inteiros 'grandes' em um sistema de computação algébrica. Por isso discutiremos este assunto brevemente. Nossa abordagem pressupõe algum conhecimento de programação. Porém, se você não tem experiência com computadores, não se preocupe: o conteúdo desta seção não é necessário para o entendimento do restante do livro.

Vamos imaginar uma situação concreta. Digamos que queremos escrever um programa para calcular com números inteiros de tamanho indeterminado. É claro que há sempre um limite superior para o tamanho máximo de um inteiro em um certo computador. Afinal de contas a área de memória do computador é finita. Quando falamos de *tamanho indeterminado* estamos querendo dizer que o computador calcula com inteiros grandes o suficiente para que, na prática, não precisemos nos preocupar com este limite máximo. Uma conseqüência disto é que não podemos representar o inteiro como um vetor (ou *array*) de tamanho pré-fixado. O sistema deve, automaticamente, aumentar ou diminuir o espaço de memória necessário para armazenar o inteiro desejado sem que precisemos interferir.

Resolvemos estes problemas representando os inteiros como *listas*. Chamaremos de b a maior potência de 10 que pode ser reprentada na linguagem de programação que estamos utilizando. O número b será, então, a *base* do sistema de numeração que usaremos para representar inteiros 'grandes' no computador. Os 'algarismos' do sistema de base b são os números de 0 a $b-1$. Representar um inteiro n na base b significa que estamos escrevendo n na forma

$$n = a_0 + a_1 \cdot b + a_2 \cdot b^2 + \cdots + a_k \cdot b^k$$

onde $a_0, a_1, \ldots, a_k$ são números entre 0 e $b-1$. Diremos que $a_0, \ldots, a_k$ são os 'algarismos' da representação de n na base b. Observe que estamos usando aspas para chamar atenção para o fato de que a palavra algarismo não está sendo usada no sentido usual. No computador, cada um destes 'algarismos' é guardado em um nó da lista que representa n. O número de nós da lista depende do tamanho do inteiro n.

O problema é que não queremos apenas representar um inteiro nesta forma: desejamos calcular com estes inteiros. A soma de dois inteiros representados como listas na base b não oferece dificuldade. Efetuamos a soma 'algarismo' a 'algarismo', exatamente como procedemos com lápis e papel, com os números representados em base 10. A única coisa que precisamos saber é quando '*vai* 1'.

Digamos que estamos somando dois inteiros positivos c e d, ambos representados na base b. É conveniente dar nomes aos algarismos de c e d, como fizemos acima para n. Chamaremos de c_0 o ' algarismo' de c correspondente à casa b^0; chamaremos de c_1 o 'algarismo' da casa b^1, e assim por diante. Analogamente, os 'algarismos' de d serão denotados por d_0, d_1, e assim por diante. Lembre-se que 'algarismo' aqui significa inteiro entre 0 e $b-1$.

Começamos a soma calculando c_0+d_0. Porém este não é necessariamente o algarismo da casa b^0 do número $c+d$. Por exemplo, se $b=10$ e $c=d=9$, então $c_0=d_0=9$ e $c_0+d_0=18$; ao passo que o algarismo das unidades é 8. O problema é que c_0+d_0 pode ser maior que a base b; neste caso o algarismo da casa b^0 de $c+d$ será c_0+d_0-b e vai 1 para a soma da casa seguinte. Observe que a regra para calcular o 'algarismo' da casa b^1 de $c+d$ é essencialmente a mesma, mas precisamos verificar se veio ou não 1 da soma dos 'algarismos' da casa anterior.

Não é difícil implementar a representação em listas, nem a adição de listas, usando *C*, *Pascal* ou outra linguagem. A multiplicação é um pouco mais complicada, embora também corresponda ao método usado com lápis e papel. A divisão, ao contrário, apresenta alguns problemas teóricos interessantes; veja a seção 2 do capítulo 1.

Uma boa parte da dificuldade de programar um sistema de computação algébrica consiste na implementação dos métodos para efetuar os cálculos com listas; mas isto não é tudo. À medida que você vai realizando cálculos, vão sendo criados inteiros que são apenas etapas necessárias para chegar ao resultado final, mas que não lhe interessam *per se*. Se você está calculando com inteiros grandes isto significa que a memória da máquina logo estará cheia, muito embora seu conteúdo não lhe interesse em nada. Por isso um sistema de computação algébrica precisa efetuar o que é conhecido pelo sugestivo nome de *coleta de lixo*. Este processo consiste em determinar quais os inteiros que podem ser eliminados e em compactar a memória disponível para uso futuro.

Com estes comentários apenas tocamos a superfície de uma vasta área, repleta de idéias, problemas e métodos próprios. Se o assunto lhe interessa,

consulte [**Knuth 1981, Cap. 4**] ou [**Davenport, Siret e Tournier 1988**].

4. Teoria de números

O estudo das propriedades dos números inteiros é atestado nas civilizações mais antigas. Entretanto é na Grécia que primeiro identificamos a teoria de números tal como a entendemos hoje em dia. No que diz respeito aos inteiros, os gregos diferenciavam entre a *logística*, ou arte de calcular com números inteiros, e a *aritmética*, ou estudo das propriedades fundamentais dos inteiros. A primeira era domínio dos comerciantes e profissionais; a segunda dos matemáticos e filósofos. A teoria dos números é herdeira da *aritmética* dos gregos. Ironicamente a palavra aritmética é usada hoje em dia para descrever aquilo que os gregos chamavam de logística.

Entre os problemas da teoria de números abordados pelos gregos antigos estão:

- o cálculo do máximo divisor comum entre dois números;
- a determinação dos números primos menores que um inteiro dado;
- a demonstração de que há uma infinidade de números primos.

Estes problemas são discutidos em detalhe num dos mais famosos livros de matemática herdados da Grécia, os *Elementos* escrito pelo matemático Euclides, que viveu em Alexandria por volta de 300 a.C..

Os Elementos são divididos em *Livros* (diríamos capítulos); treze no total. Três desses livros lidam com problemas da teoria de números; os demais tratam de questões ligadas aos números reais e à geometria. O primeiro dos livros a tratar da teoria de números é o VII. Nele encontramos as definições de número primo e o método para achar o máximo divisor comum entre dois números, que estudaremos no capítulo 1. O *Livro VIII* trata principalmente das propriedades das progressões geométricas. Finalmente o *Livro IX* contém a demonstração de que há infinitos números primos e se encerra com o estudo dos números perfeitos. Estudaremos a demonstração de Euclides da infinidade dos primos no capítulo 3; os números perfeitos aparecerão nos exercícios do capítulo 2.

Vários outros matemáticos gregos estudaram problemas da teoria de números. Destes o mais importante foi sem dúvida Diofanto. Sua *Aritmética*, escrita por volta de 250 d.C., trata principalmente da solução de equações indeterminadas com coeficientes inteiros. Trataremos de questões deste tipo no capítulo 4. Embora a matemática tenha sido intensamente estudada por outros autores gregos, e posteriormente árabes, indianos e europeus, a teoria

de números caiu mais ou menos no esquecimento até o século XVII.

5. Fermat, Euler e Gauss

Durante a Renascença muitas obras de autores gregos foram redescobertas, editadas e publicadas na Europa. Entretanto isto não aconteceu com a maior parte dos matemáticos gregos. Por exemplo, foi somente em 1621 que Bachet publicou o texto original em grego da *Aritmética* de Diofanto, juntamente com uma tradução latina, que era a língua usada pelos eruditos europeus da época. Em algum momento entre 1621 e 1636 o francês Pierre de Fermat, magistrado da corte de Toulouse, adquiriu uma cópia deste livro. Fermat era um matemático amador (mas somente no sentido de que não vivia da matemática). Nas horas vagas, leu o texto de Diofanto, anotando na margem as idéias que lhe ocorriam. Isto marcou o início do interesse de Fermat em teoria de números, que posteriormente se expressou em uma torrente de resultados importantes, muitos dos quais estudaremos neste livro.

P. de Fermat (1601–1665).

Fermat nasceu em 1601, e como já vimos era um magistrado por profissão, e não matemático. Na verdade poucas pessoas viviam de matemática naquela época. A comunicação entre os matemáticos também era precária. Não havia revistas especializadas; a primeira revista dedicada à matemática só foi criada em 1794. A comunicação era conduzida principalmente através de cartas, com algumas pessoas servindo de centros difusores dos novos resultados. O mais famoso dentre estes foi o frade francês Marin Mersenne.

Muito amigo de alguns dos maiores matemáticos da época, como Descartes, Pascal e o próprio Fermat, Mersenne logo comunicava a toda a 'República de Letras' as novidades matemáticas que chegavam ao seu conhecimento.

Foi na forma de cartas enviadas a Mersenne e a outros matemáticos contemporâneos que boa parte da obra de Fermat chegou até nós. Depois da morte de Fermat em 1665, coube a Samuel Fermat coletar e publicar a obra de seu pai, dispersa em cartas e anotações. Ele começou com a publicação da *Aritmética* de Diofanto, incluindo todas as anotações feitas por Fermat à margem da sua cópia. Destas anotações a mais famosa é o chamado *Último Teorema de Fermat*: se x, y e z são inteiros e

$$x^n + y^n = z^n,$$

onde $n \geqslant 3$, então $xyz = 0$. Este resultado só foi provado em 1995–quase 300 anos depois de ser anunciado por Fermat. Para mais detalhes veja capítulo 2, seção 8.

A obra matemática de Fermat não se limita à teoria de números. Ele deu contribuções importantes ao cálculo diferencial e integral, à geometria e à teoria de probabilidade, por exemplo. Entretanto foi Fermat quem redescobriu a teoria de números, diretamente da fonte grega, e divulgou esta área na Europa do século XVII. Infelizmente Fermat nunca chegou a interessar nenhum de seus contemporâneos neste assunto. O verdadeiro sucessor de Fermat foi o suíço Leonhard Euler, que nasceu em 1707, quarenta e dois anos depois da morte de Fermat.

Euler publicou uma obra matemática imensa, tendo contribuído para quase todas as áreas da matemática pura e aplicada existentes no século XVIII. Ao contrário de Fermat, a matemática era a principal ocupação profissional de Euler. Ele não foi professor de nenhuma universidade, mas esteve ligado a academias científicas na Alemanha e na Rússia. Essas academias eram, na verdade, instituições de pesquisa, cujas atas publicavam primordialmente as contribuições científicas de seus membros.

O interesse de Euler em teoria de números teve início em sua correspondência com Christian Goldbach. Como Mersenne no século anterior, Goldbach não era um grande matemático, mas alguém que tinha na matemática um 'hobby'. Entretanto foi através dele que Euler chegou à obra de Fermat. Em sua primeira carta a Euler em 1729, Goldbach acrescenta o seguinte PS: *Você conhece a observação de Fermat de que todos os números* $2^{2^n} + 1$ *são primos? Ele disse que não sabia prová-la; nem ninguém conseguiu fazê-lo, que eu tenha conhecimento.*

L. Euler (1707–1783).

Euler reage com ceticismo e não demonstra muito interesse, mas Goldbach não desiste e volta ao assunto. Em 1730 Euler começa finalmente a ler a obra de Fermat. Nos anos seguintes ele provaria e estenderia grande parte dos resultados enunciados por Fermat, resolvendo inclusive a questão proposta por Goldbach, como veremos no capítulo 9.

Através de seus trabalhos Euler popularizou a teoria de números como Fermat não o havia conseguido. Mas o desenvolvimento sistemático da teoria só teria início com as *Disquisitiones arithmeticæ* do alemão C.F. Gauss publicada em 1801. O enfoque atual da teoria, inclusive o que é adotado neste livro, teve origem na obra de Gauss. Muitos dos resultados que veremos adiante vêm diretamente das *Disquisitiones*. Gauss era filho de um trabalhador braçal, mas foi uma criança prodígio e sua habilidade matemática foi notada muito cedo. Suas contribuições à teoria de números, geometria diferencial, análise e física são tantas que ele ficou conhecido como o 'príncipe dos matemáticos'.

Fermat, Euler e Gauss são os verdadeiros heróis deste livro. A maior parte das idéias que estudaremos são originárias ou da Grécia Antiga, ou de um destes três matemáticos.

Não podemos encerrar nossa discussão da teoria de números sem tentar explicar o que a faz tão cativante que Gauss a chamou de 'rainha da matemática'. A melhor maneira de dar uma idéia do que há de atraente em uma área da matemática é descrever alguns dos seus problemas. Os problemas da teoria de números se caracterizam pela simplicidade do enunciado, mas

enorme dificuldade das demonstrações, que exigem freqüentemente grande engenhosidade e virtuosidade técnica. A seguir relacionamos alguns exemplos. Depois de lê-los todos, faça uma pausa e tente advinhar quais os mais difíceis e quais os mais fáceis.

(1) Todo número da forma $2^{2^n} + 1$, com $n \geqslant 0$, é primo?
(2) Todo número inteiro par maior que 2 é soma de dois primos?
(3) Se p é um primo qualquer então p divide $2^{p-1} - 1$?
(4) Existe algum primo p tal que p^3 divide $2^{p-1} - 1$?
(5) Existem dois inteiros consecutivos, além de 8 e 9, que são potências (de inteiros)?
(6) Todo primo ímpar pode ser escrito como a soma de dois quadrados de inteiros?
(7) Existem infinitos primos p para os quais $p + 2$ também é primo?

Como já vimos o problema (1) foi proposto por Fermat e comunicado por Goldbach a Euler, que mostrou que $2^{2^5} + 1$ é composto. Estudaremos no capítulo 9 o método pelo qual Euler chegou a um fator. O problema (2) é conhecido como *conjectura de Goldbach* e até hoje não se sabe se a afirmação é verdadeira ou falsa.

Os problemas (3) e (4) são muito semelhantes. Comparando-os podemos até achar que (4) deve ser mais fácil de resolver que (3). Afinal (4) pergunta se *existe algum primo* satisfazendo uma certa propriedade, ao passo que (3) pergunta se uma propriedade muito semelhante vale *para todos os primos*. A verdade é que (3) é um teorema demonstrado por Fermat no século XVII e que estudaremos no capítulo 5. Já (4) é uma pergunta para a qual ninguém conhece a resposta.

O problema (5) é conhecido como *conjectura de Catalan*, e também não se sabe se é verdadeiro ou falso. Sabe-se, porém, que se a diferença entre um número inteiro ao cubo e um outro inteiro ao quadrado for ± 1 então o cubo é 8 e o quadrado é 9. Isto foi provado por Euler, e a demonstração não é muito difícil. Para mais detalhes sobre a conjectura de Catalan, veja [**Ribenboim 1996**].

A resposta do problema (6) é não. Se um primo ímpar deixa resto 3 quando dividido por 4, então não é possível escrevê-lo na forma $x^2 + y^2$, com x e y inteiros. Isto foi provado por Fermat e você encontrará a demonstração no exercício 10 do capítulo 4. Contudo os números primos ímpares que deixam resto 1 quando divididos por 4 *podem* ser escritos como somas de quadrados de inteiros. Isto também foi mostrado por Fermat. Para mais detalhes veja

o exercício 12 do capítulo 5. A história destes problemas é contada em [**Weil 1987, Cap. II, §VIII**].

Finalmente o problema (7) é a famosa *conjectura dos primos gêmeos*, que também não se sabe se é verdadeira ou falsa. O que se sabe é que existem infinitos números primos. Isto é fácil de mostrar e já era conhecido de Euclides; veremos a demonstração na seção 5 do capítulo 3. Sabe-se também que, se a e r são inteiros positivos cujo máximo divisor comum é 1, então a progressão aritmética de primeiro termo a e razão r contém uma infinidade de primos. Um caso particular deste resultado é provado nos exercícios 3 a 7 do capítulo 3. Uma outra conjectura relacionada à dos primos gêmeos pergunta se existem infinitos primos p para os quais $p + 2$ e $p + 6$ também são primos. Mas observe que só há um primo p para o qual $p + 2$ e $p + 4$ também são primos; veja o exercício 9 do capítulo 3.

6. O livro

Como já dissemos, o objetivo deste livro é estudar a matemática que serve de fundamento ao método de criptografia RSA. A maior parte do que precisamos será encontrado nos métodos da teoria de números desenvolvidos pelo gregos antigos e pelos matemáticos Fermat, Euler e Gauss, entre os séculos XVII e XIX. Se os fundamentos da teoria já têm quase 200 anos, muitas das aplicações que nos interessam têm menos de 20 anos. Alguns dos resultados que mencionaremos foram publicados há apenas dois ou três anos.

Muito do conteúdo deste livro já lhe é conhecido desde o primeiro grau. Este certamente é o caso da divisão de inteiros, do máximo divisor comum, da fatoração de inteiros e dos números primos. É no enfoque que provavelmente estará a novidade, já que nos propomos a provar rigorosamente todos os resultados que estudaremos.

Para os egípcios e mesopotâmicos a matemática era uma coleção de regras para resolver problemas práticos. Segundo o historiador grego Heródoto, a geometria teria sido inventada no Egito por causa das enchentes do Nilo. As inundações ocupavam as terras férteis e mudavam um pouco o leito do rio. Ao cabo de cada inundação seria necessário redistribuir os terrenos de modo que cada agricultor mantivesse a mesma área de cultivo.

Foi a influência da filosofia grega que fez da matemática uma ciência. De fato os primeiros matemáticos gregos estão entre os primeiros filósofos, como é o caso de Tales e Pitágoras. A noção de que um *fato matemático* pode ser *demonstrado* é fruto da interação entre matemática e filosofia. Afinal, uma demonstração é essencialmente um argumento para esclarecer como

um certo fato é conseqüência de algo que já conhecemos. E se há alguma coisa que os filósofos gregos gostavam de fazer era argumentar.

Só posteriormente as hipóteses fundamentais passaram a ser enunciadas de maneira sistemática. Assim *Os Elementos* de Euclides começa com definições e axiomas. Por exemplo, no *Livro I* são definidos ponto, reta, plano, superfície e assim por diante. A seguir são enunciados axiomas, ou verdades primitivas, que descrevem como se relacionam os conceitos definidos anteriormente; por exemplo, por dois pontos passa uma e apenas uma reta. A partir destas definições e axiomas, Euclides vai argumentando, passo a passo, de modo a deduzir resultados mais complexos.

Observe que mesmo na Grécia conviviam lado a lado a *matemática prática* dos comerciantes e artesãos e a *matemática teórica* dos filósofos. Assim, como já vimos, havia a *logística*, que tratava do cálculo com inteiros, e a *aritmética*, que é a nossa teoria de números. É muito provável que sua atitude atual em relação aos números inteiros corresponda à logística dos gregos. Ao ler este livro você vai precisar mudar seu enfoque e começar a pensar em termos de teoria dos números, propriedades fundamentais e demonstrações.

Aí está a principal dificuldade deste livro. Não se trata de matemática nova, mas de uma maneira diferente de abordar muitas coisas que até já lhe são conhecidas. Infelizmente mudar atitudes é justamente o mais difícil. Mas não se intimide; apenas prepare-se para a novidade. Ao invés de descrever uma fórmula e começar a aplicá-la, estaremos dedicando a maior parte de nosso tempo a entender porque a fórmula é verdadeira. Isto vai dar maior robustez àquilo que você for aprendendo, permitindo assim estudar questões muito mais delicadas e interessantes.

7. Teoremas e demonstrações

Um fato matemático é freqüentemente chamado de *teorema*. A palavra vem do grego e originalmente significava 'espetáculo ou festa'. O sentido moderno de 'proposição a ser demonstrada' é atestado a partir dos *Elementos* de Euclides. O enunciado de um teorema é constituído de duas partes, as *hipóteses* e a *tese*. As hipóteses descrevem aquilo que estamos supondo ser verdade; a tese é a conclusão do teorema. Vejamos um exemplo.

Teorema 1. *Se* a *é um número inteiro par então* a^2 *também é par.*

A hipótese neste caso é a *é inteiro par*, a tese é a^2 *é par*. Um teorema é freqüentemente enunciado na forma: se *hipótese* então *tese*; como é o caso acima. A *demonstração* do teorema é um argumento lógico que nos permite

partir da hipótese e chegar à tese. Vejamos como demonstrar o teorema acima.

Demonstração do teorema 1: A hipótese nos diz que a é inteiro par. Isto significa, por definição, que a é múltiplo de 2. Portanto deve existir um inteiro b tal que $a = 2b$. Elevando a ao quadrado obtemos

$$a^2 = (2b)^2 = 4b^2 = 2 \cdot (2b^2).$$

Assim a^2 também é múltiplo de 2; logo a^2 é par. Mas isto é a tese, e portanto o teorema está provado.

Vimos no teorema acima que o fato de a ser par *implica* que a^2 também seja par. A *recíproca* de uma afirmação do tipo 'A implica B' é a afirmação 'B implica A'. Assim a recíproca do teorema acima é: se a^2 é um inteiro par então a é um inteiro par. Não há nenhuma garantia de que a recíproca de uma afirmação verdadeira também seja verdadeira. Por exemplo, a recíproca de 'se um número inteiro é múltiplo de 4 então é par' é falsa: 2 é par mas não é múltiplo de 4. Normalmente a afirmação 'valem A implica B e B implica A' é abreviada na forma 'A vale se e somente se B vale'. Chegamos assim ao nosso segundo teorema.

Teorema 2. *a é um inteiro par se, e somente se, a^2 é um inteiro par.*

Já provamos que se a é par então a^2 é par. Falta demonstrar a recíproca. Antes precisamos esclarecer um ponto de lógica. Imagine que um amigo lhe diga: 'eu vou à festa se chegar cedo em casa'. À noite você espera seu amigo na festa e ele não vem. Você imediatamente conclui que ele não chegou cedo em casa. Vamos analisar a lógica por trás da sua dedução. Podemos dissecar a afirmação de seu amigo na seguinte forma: seja P a afirmação 'chego cedo' e Q a afirmação 'vou à festa'. O que seu amigo lhe disse equivale a 'P implica Q'. Disto você deduziu 'não veio à festa então não chegou cedo'. Digamos que 'não-P' corresponde à negação de P; no exemplo, 'não-P' é 'não chego cedo'. A sua dedução corresponde a 'não-Q implica não-P'. Observe que 'P implica Q' é verdadeira exatamente quando 'não-Q implica não-P' é verdadeira. Dizemos que uma destas afirmações é a *contrapositiva* da outra. Moral: tanto faz provar uma afirmação quanto a sua contrapositiva, já que são equivalentes. Voltemos à demonstração do teorema 2.

Demonstração do teorema 2: Já mostramos que se a é par então a^2 é par. Falta mostrar que se a^2 é par então a é par. Na verdade o que vamos provar é a contrapositiva desta afirmação, que é: se a não é par então a^2 não é par. Mas se um número inteiro não é par, então tem que ser ímpar. Um número

ímpar é sempre da forma 'par + 1'. Logo, se a é ímpar, tem que existir um inteiro b tal que $a = 2b + 1$. Elevando a ao quadrado obtemos

$$a^2 = (2b + 1)^2 = 4b^2 + 4b + 1 = 2 \cdot (2b^2 + 2b) + 1$$

que é um número ímpar. Portanto a contrapositiva da afirmação que queremos provar é verdadeira. Como contrapositiva e afirmação original são equivalentes, provamos que se a^2 é par então a é par.

Enunciamos o teorema 1 na forma: se a é par então a^2 é par. Observe que o que isto quer dizer é: se a é um número par *qualquer* então a^2 é par. Isto é, estamos afirmando que o fato enunciado vale para *todo* número par. Considere esta outra afirmação: todo número par é múltiplo de 4. Temos uma nova afirmação a respeito de todos os números pares; mas desta vez a afirmação é falsa. Por quê? Porque 2, por exemplo, é par mas não é múltiplo de 4. Conclusão: imagine que é dada uma afirmação que diz que uma certa propriedade vale para qualquer elemento de um certo conjunto. Se houver um elemento do conjunto para o qual a proposição *não* vale então a proposição como um todo é falsa. Dizemos que um elemento que falsifica uma proposição é um *contra-exemplo* à sua validade.

Estas anotações são suficientes para você começar a ler o livro. À medida que formos avançando, introduziremos vários métodos de demonstração para poder alcançar resultados mais sofisticados. Note porém que não existe um método universal de demonstração, que possa ser aplicado de forma mecânica. Como qualquer habilidade, a arte de demonstrar tem que ser cultivada. Cultivar uma arte significa praticá-la com assiduidade. Quando Ptolomeu, o rei do Egito, perguntou a Euclides se não havia uma maneira mais fácil de aprender geometria que ler os treze livros dos *Elementos*, Euclides respondeu *'em geometria, não existe uma estrada especial para os reis'*–o que era verdade no tempo de Euclides continua valendo hoje em dia.

8. Pré-requisitos

O principal pré-requisito para a leitura deste livro é um certo grau de maturidade matemática. Seria muito bom que nisto estivesse incluída alguma familiaridade com a arte de escrever demonstrações. Do ponto de vista técnico, é essencial ter proficiência no cálculo com potências e na manipulação de expressões algébricas (literais). Do segundo grau os tópicos que usaremos mais freqüentemente são o binômio de Newton e as progressões geométricas (sobretudo a fórmula da soma).

1

Algoritmos fundamentais

São dois: o algoritmo de divisão e o algoritmo euclidiano. Ambos eram conhecidos na Grécia Antiga e são descritos nos *Elementos* de Euclides, escrito por volta de 300 a.C.. O algoritmo de divisão que nos interessa é o que calcula quociente e resto na divisão de um número inteiro por outro. O algoritmo euclidiano é usado para calcular o máximo divisor comum. Que são fundamentais, você vai se convencer à medida que for lendo este livro.

1. Algoritmos

O Aurélio define a palavra *algoritmo* da seguinte maneira:
processo de cálculo, ou de resolução de um grupo de problemas semelhantes, em que se estipulam, com generalidade e sem restrições, regras formais para a obtenção de um resultado, ou da solução de um problema.
Como a maior parte das definições dos dicionários, esta não ajuda muito a identificar um algoritmo, se você se deparar com um. Na verdade um algoritmo é, essencialmente, uma *receita* para resolver um determinado tipo de problema.

Para ajudá-lo a entender como um algoritmo é apresentado, vamos analisar com cuidado uma receita simples. Nossa tarefa é fazer um bolo. Um bom livro de receitas tem, logo depois do nome do bolo, uma lista dos *ingredientes* necessários à sua confecção. Seguem-se as instruções do que fazer com os ingredientes para chegar ao objetivo; coisas como: peneire, misture, bata, asse. Finalmente, temos o resultado: o bolo pronto.

Assim é todo algoritmo. Quando formos especificar o algoritmo, deveremos deixar claro qual é a sua *entrada* e a sua *saída*. A entrada corresponde aos ingredientes da receita. A saída corresponde à tarefa que desejamos ver executada; no caso acima, o bolo que desejamos fazer. O algoritmo propriamente dito é constituído pelo conjunto de instruções da receita.

Tendo seguido uma receita de bolo com o devido cuidado, esperamos, ao

tirar o assado do forno, encontrar um bolo, e não um biscoito ou um bife de panela. Também esperamos que a receita possa ser acabada em um tempo finito, de preferência curto. Da mesma forma, tendo executado o algoritmo, esperamos obter um resultado compatível com a saída especificada. E mais, esperamos que o algoritmo chegue à saída ao cabo de um tempo finito, de preferência pequeno. Assim há dois pontos a considerar. Em primeiro lugar, é possível escrever uma seqüência de instruções que continua sendo executada para sempre. Por exemplo, dado um número inteiro qualquer (a entrada) some 1 a este número, depois some 1 ao resultado e assim por diante. Como existem infinitos inteiros, o processo continua para sempre. Como esperamos que um algoritmo forneça um resultado, um conjunto de instruções deste tipo não tem nenhuma utilidade.

Por outro lado, um algoritmo pode ser lento e mesmo assim ser útil. Talvez não haja um algoritmo melhor; ou talvez o algoritmo sirva para mostrar que o problema tem solução. É claro que nem todo problema pode ser resolvido pela execução de um conjunto de instruções! Menos claro é que nem todo problema *matemático* pode ser resolvido desta forma–mas esta é uma outra história, longa demais para contar aqui. Para mais detalhes sobre problemas que não podem ser resolvidos usando algoritmos veja [**Davis 1980**].

Podemos enunciar um algoritmo como um fato ou teorema: dada tal entrada existe uma maneira (o algoritmo) de produzir tal saída. Um teorema normalmente é enunciado na forma '*se valem as hipóteses então temos a tese*'. No caso do teorema associado a um algoritmo, a entrada do algoritmo corresponde às hipóteses do teorema, e a saída à tese.

Não se preocupe se estes comentários lhe parecem muito vagos, estamos apenas estabelecendo a nomenclatura. À medida que os exemplos forem surgindo tudo vai ficar mais claro. Resumindo o que já dissemos: um algoritmo é uma receita, um conjunto de instruções que toma certos ingredientes (a entrada) e transforma num certo produto (a saída). Digamos que alguém nos apresenta um conjunto de instruções e afirma que se trata de um algoritmo para executar determinada tarefa. Somos, então, moralmente obrigados a fazer duas perguntas:

- *ao executar estas instruções, sempre chegaremos a um resultado?*
- *o resultado obtido é sempre o desejado?*

Com isto em mente, faremos uma análise cuidadosa do *algoritmo de divisão*.

A etimologia da palavra algoritmo é tão curiosa que seria uma pena encerrar a seção sem mencioná-la. Originalmente a palavra era escrita *algorismo*, que vem da forma latinizada do árabe Al-Khowarazmi, 'o homem de Kho-

warazm'. Assim era conhecido o matemático árabe Ben Musa, que viveu no século IX. Foi através de seu livro 'Al-jabr wa'l muqabalah' que os algarismos indo-arábicos chegaram ao Ocidente. Aliás *algorismo* e *algarismo* são variantes da mesma palavra e significavam originalmente os numerais indo-arábicos. Por extensão algorismo era também usado para significar aritmética e cálculo com números. Algoritmo ganhou um 't' por influência da palavra '*aritmos*'–que também significa número, só que em grego.

O sentido atual da palavra algoritmo, contudo, é bem mais recente, sendo atestado, em inglês, apenas a partir de 1812. Não é claro como a palavra chegou a ter este sentido. É possível testemunhá-la adquirindo um significado mais abrangente no artigo de 1684 em que Leibniz apresenta, pela primeira vez, sua versão do cálculo diferencial e integral. No começo deste artigo Leibniz diz que as regras do cálculo que vai apresentar podem ser consideradas como um 'algoritmo'. Em pouco mais de um século a palavra já havia praticamente tomado o sentido atual. Gauss, escrevendo em latim, usa 'algoritmus' repetidas vezes em suas *Disquisitiones Arithmeticæ* quando se refere a um conjunto de fórmulas que constitui um método para calcular a solução de algum problema aritmético.

Assim algoritmo é uma palavra muito antiga, mas que ganhou um significado novo. Em tempo: do nome do livro de Ben Musa, mencionado acima, vem a nossa palavra *álgebra*.

2. Algoritmo de divisão

Vamos analisar o algoritmo de divisão na forma estabelecida na seção anterior. Em primeiro lugar, só estamos interessados na divisão de um inteiro por outro, a *divisão com resto*. Portanto, nossa tarefa é achar o quociente e o resto da divisão entre dois inteiros positivos. Ao ouvir isto você imaginará certamente algo assim

$$\begin{array}{r|l} 1234 & 54 \\ \cline{2-2} 154 & 22 \\ 46 & \end{array}$$

Neste exemplo estamos dividindo 1234 por 54 e encontramos o quociente 22 e o resto 46. Na linguagem da seção anterior, o algoritmo tem por entrada o dividendo e o divisor–neste caso 1234 e 54. A saída são o quociente e o resto–neste caso 22 e 46.

De uma maneira geral, a *entrada* do algoritmo de divisão são os dois inteiros positivos a e b, que desejamos dividir. A *saída* são outros dois inteiros

q e r, que estão relacionados à entrada de maneira que

$$a = bq + r \qquad \text{e} \qquad 0 \leqslant r < b,$$

como sabemos desde o primário. Falando em primário, é bom relembrar a interpretação do algoritmo que aprendemos naquela época. Imagine que temos uma barra de chocolate de comprimento a e que queremos obter o maior número possível de pedaços de comprimento b dividindo a barra. A resposta é que obteremos q pedaços e uma 'sobra' que terá comprimento r.

A barra de chocolate é a inspiração para o algoritmo mais simples possível para achar q e r, a partir de a e b. Simples, mas não eficiente. Na verdade trata-se de um algoritmo altamente ineficiente. Podemos descrevê-lo assim.

Algoritmo de divisão.
Entrada: *inteiros positivos* a *e* b.
Saída: *inteiros não-negativos* q *e* r *tais que* $a = bq + r$ *e* $0 \leqslant r < b$.

Etapa 1: *Comece fazendo* $Q = 0$ *e* $R = a$.
Etapa 2: *Se* $R < b$ *escreva o quociente é* Q *e o resto é* R *e pare; senão vá para a Etapa 3.*
Etapa 3: *Se* $R \geqslant b$ *subtraia* b *de* R, *incremente* Q *de* 1 *e volte à Etapa 2.*

Ao longo deste livro os algoritmos serão freqüentemente estruturados na forma descrita acima. A leitura correta das instruções pressupõe algumas convenções. Observe que o algoritmo usa duas *variáveis*, Q e R. Os nomes das variáveis foram escolhidos por razões mnemônicas: ao final da execução do algoritmo Q vai ser o valor do quociente e R o valor do resto na divisão de a por b. Para determinar estes valores teremos geralmente que executar as instruções das Etapas 2 e 3 várias vezes: chamamos a isto um *laço*. Ao final de cada laço as variáveis Q e R terão valores diferentes. A mudança de valor é efetuada na Etapa 3. A instrução *subtraia* b *de* R significa: à variável R será atribuído um novo valor que corresponde ao valor que tinha ao final do laço anterior menos b. Analogamente *incremente* Q *de* 1 significa: à variável Q será atribuído o valor que tinha ao final do laço anterior mais 1.

Por exemplo, se $a > b$, então tendo passado uma vez pela Etapa 3 obtivemos $Q = 1$ e $R = a - b$. Se $a - b \geqslant b$ então o algoritmo nos manda aplicar a Etapa 3 mais uma vez. Tendo feito isto, teremos obtido $Q = 2$ e $R = a - 2b$. E assim por diante. Por que isto não se repete para sempre? Ou, em outras palavras, por que é que o algoritmo pára? Observe que aplicando a Etapa 3

várias vezes geramos a seguinte seqüência de valores da variável R:

valor inicial	1º laço	2º laço	3º laço	...
a	$a-b$	$a-2b$	$a-3b$	...

Trata-se de uma seqüência decrescente de números inteiros. Como existe apenas uma quantidade finita de inteiros entre a e 0, a seqüência acima vai chegar eventualmente a um valor *menor* que b. Neste momento a Etapa 2 nos manda encerrar o processo e divulgar a resposta. Logo o algoritmo sempre pára.

Que o resultado da aplicação do algoritmo corresponde às especificações da saída é fácil de ver. Note que se q e r são obtidos pelo algoritmo então, automaticamente, temos $r = a - bq$ e $r < b$. Da primeira equação segue facilmente que $a = bq + r$. Falta apenas descobrir porque $r \geqslant 0$. Pelo que vimos acima, o algoritmo parou no laço de número q. Logo no laço anterior (que é $q - 1$) temos $a - b(q-1) \geqslant b$ ou não estaríamos subtraindo b mais uma vez para chegar ao laço q. Subtraindo b de ambos os membros desta última equação chegamos a $a - bq \geqslant 0$, que é o que queríamos verificar. Logo o algoritmo descrito realiza o que dele se pede.

Não é preciso pensar muito para ver que o algoritmo apresentado é muito lento. O número de laços executados é igual ao quociente; assim, se a for muitas ordens de grandeza maior que b, estamos perdidos. O método usual de divisão oferece uma maneira prática de acelerar este algoritmo. Infelizmente este método (como costumamos usá-lo) é excelente para seres humanos dividirem números pequenos e péssimo quando os números são grandes ou o calculista é uma máquina.

Para entender o porquê da observação acima vamos acompanhar os cálculos na divisão do começo da seção. Em primeiro lugar escolhemos o menor número formado pelos algarismos de 1234 (começando da esquerda) que é maior que 54. Este número é 123. Fazemos, então, a pergunta: *'qual o maior inteiro que posso multiplicar por* 54 *e ainda assim obter um produto menor que* 123*?'* E é nesta pergunta que reside o problema. Se os números são pequenos, um pouco de experiência e algumas tentativas nos levam rapidamente ao valor correto. Se os números são grandes, nossa experiência é inútil, e a quantidade de possibilidades pode tornar impossível atacar o problema por tentativa. Há uma saída para este impasse, mas discuti-la aqui nos levaria a uma digressão muito longa. Para mais detalhes sobre esta questão consulte o livro [**Knuth 1981,** §**4.3**].

Antes de encerrar a seção precisamos considerar um problema prático. A maior parte dos resultados abordados neste livro só se torna interessante se

os números a que se aplicam são 'relativamente grandes'; grandes o bastante para tornar o cálculo de divisões com lápis e papel impraticável. Isto sem falar no fato de que teremos de executar dezenas de divisões na solução de alguns problemas. A saída é usar uma máquina de calcular. Mas máquina de calcular não faz divisão com resto–ou faz? Imagine que temos dois inteiros positivos a e b e desejamos calcular a expressão decimal da fração a/b. O começo da divisão é exatamente igual ao da divisão com resto. Só que ao invés de parar e escrever o resto, colocamos a vírgula decimal e continuamos a dividir. Moral: a parte inteira da expressão decimal de a/b é o *quociente* da divisão com resto de a por b. Se o quociente é q então o resto é simplesmente $a - bq$; e achamos quociente e resto com uma calculadora. Se a calculadora tem memória é muito fácil executar esta conta (com um pouco de prática) em tempo recorde.

Mas cuidado: o tamanho dos inteiros a serem divididos está limitado pelo número máximo de algarismos que o visor da calculadora aceita (normalmente oito ou dez). Talvez você esteja pensando que isto não é um grande problema. Afinal sua calculadora escreverá números de até 99 algarismos em *notação científica*; isto é na forma $r \cdot 10^n$, onde $1 \leqslant r < 10$. O problema é que precisaremos conhecer o quociente exatamente. Mas se sua calculadora está usando notação científica isto normalmente significa que a parte inteira do quociente está sendo truncada.

3. Teorema de divisão

Na seção 1 também mencionamos que todo algoritmo vem acompanhado de um teorema. Vamos enunciar o teorema correspondente ao algoritmo de divisão.

Teorema de divisão. *Sejam a e b inteiros positivos. Existem números inteiros q e r tais que*

$$a = bq + r \quad e \quad 0 \leqslant r < b.$$

Além disso, os valores de q e r satisfazendo as relações acima são únicos.

No enunciado acima, o que vem antes do 'além disso' não oferece surpresa. Que, dados a e b, existem q e r já sabemos. Sabemos até como calculá-los usando o algoritmo descrito acima. O que vem depois do 'além disso' é que é a novidade. O que quer dizer que os valores são únicos? Imagine que escolhemos dois inteiros positivos a e b e os distribuímos a várias pessoas com a instrução de que determinem (do jeito que bem desejarem) inteiros q e r satisfazendo as relações do teorema. A unicidade do quociente e do

resto significa que *todas estas pessoas encontrarão os mesmos valores*. Em particular algoritmos distintos para calcular quociente e resto *não* podem achar valores distintos, o que é uma informação muito útil.

Vejamos porque isto é verdade. Digamos que escolhemos dois inteiros positivos a e b. Entregamos estes números a duas pessoas, Carlos e Sofia, e pedimos que determinem quociente e resto satisfazendo as condições do teorema. Carlos encontrou q e r e Sofia q' e r'. Sabendo apenas que

$$a = bq + r \quad \text{e} \quad 0 \leqslant r < b$$

e que

$$a = bq' + r' \quad \text{e} \quad 0 \leqslant r' < b$$

mostraremos que $q = q'$ e $r = r'$. Lembre-se que r e r' são inteiros; logo um deles é maior ou igual ao outro. Para fixar as idéias, digamos que $r \geqslant r'$. Da equação de Carlos $r = a - bq$ e da equação de Sofia $r' = a - bq'$. Subtraindo uma da outra

$$r - r' = (a - bq) - (a - bq') = b(q' - q).$$

Por outro lado, tanto r quanto r' são menores que b. Como estamos supondo também que $r \geqslant r'$, obtemos $0 \leqslant r - r' < b$. Comparando com $r - r' = b(q' - q)$ concluímos que

$$0 \leqslant b(q' - q) < b.$$

Mas b é um inteiro positivo e pode ser cancelado, logo $0 \leqslant q' - q < 1$. Como $q' - q$ é um número inteiro, estas desigualdades só se verificam quando $q' - q = 0$, ou seja $q = q'$. Disto segue imediatamente que $r = r'$, e verificamos a unicidade.

Recapitulando, vimos que o algoritmo de divisão dá lugar a um teorema que nos diz *duas* coisas: que o quociente e o resto sempre *existem*, e que o quociente e o resto são *únicos*. Muitos outros teoremas que estudaremos neste curso afirmarão existência e unicidade de certas condições. O mais importante deles será o *teorema da fatoração única* a ser estudado no próximo capítulo.

4. Algoritmo euclidiano

O objetivo do algoritmo euclidiano é calcular o máximo divisor comum entre dois números inteiros. Mesmo correndo o risco de parecer pedante, vamos definir cuidadosamente o que deve ser entendido por máximo divisor comum.

Em primeiro lugar um inteiro b *divide* outro inteiro a se existe um outro número inteiro c tal que $a = bc$. Também dizemos que b é um *divisor* ou

fator de a, ou ainda que a é *múltiplo* de b. Todas estas expressões significam a mesma coisa. O número c, na definição acima é chamado de *co-fator* de b em a. Na prática, determinamos que b divide a efetuando a divisão e verificando que o resto é zero. O co-fator é o quociente da divisão.

Suponhamos que temos dois inteiros positivos a e b. O *máximo divisor comum* entre a e b é o *maior* inteiro positivo d que é divisor de a e também é divisor de b. Se d é o máximo divisor comum entre a e b, escrevemos $d = \mathrm{mdc}(a, b)$. Se $\mathrm{mdc}(a, b) = 1$, dizemos que os números são *primos entre si* ou *co-primos*.

Um livro da quarta série primária descreve a seguinte maneira de calcular o máximo divisor comum entre a e b. Primeiro calcule *todos* os divisores de a; depois *todos* os divisores de b. Foram obtidos dois conjuntos: calcule a intersecção dos dois. O maior número inteiro contido na intersecção é o máximo divisor comum desejado. O que há de errado nisto? Nada; mas se precisássemos recorrer a este método para calcular o máximo divisor comum, nunca chegaríamos a usá-lo para nada. Infelizmente, achar todos os divisores de um inteiro é *totalmente impraticável* quando o número é grande, mesmo usando um bom computador. Discutiremos isto com detalhes no próximo capítulo. Milagrosamente existe uma outra maneira de calcular o máximo divisor comum através de um algoritmo muito eficiente. Este algoritmo é descrito por Euclides nas proposições 1 e 2 do Livro 7 dos *Elementos*, mas acredita-se que sua origem seja muito anterior a Euclides.

Digamos, mais uma vez, que a e b sejam inteiros positivos e que $a \geqslant b$. Queremos calcular o máximo divisor comum entre a e b. O *algoritmo euclidiano* consiste em dividir a por b, achando o resto r_1. Se $r_1 \neq 0$, dividimos b por r_1, obtendo o resto r_2. Se $r_2 \neq 0$, dividimos r_1 por r_2, obtendo o resto r_3. E assim por diante. O último resto *diferente de zero* desta seqüência de divisões é o máximo divisor comum entre a e b. Quando o cálculo do máximo divisor comum é feito com lápis e papel, é costume arrumar as divisões em um quadro. No exemplo abaixo, calculamos o máximo divisor comum entre 1234 e 54.

1234	54	46	8	6	2
46	8	6	2	0	

No quadro acima escrevemos apenas os restos das divisões. Os quocientes não são usados diretamente no cálculo do máximo divisor comum. De acordo com a regra exposta no parágrafo anterior, temos que $\mathrm{mdc}(1234, 54) = 2$, que é o último resto não nulo da seqüência de divisões. Temos então o al-

goritmo: para calcular o máximo divisor comum tudo o que temos a fazer é calcular uma seqüência de divisões. Mas porquê o resultado destas divisões é o máximo divisor comum? Além disso, precisamos verificar que a seqüência de divisões chega sempre a um resto zero, ou o algoritmo continuaria para sempre.

Começaremos tratando a segunda questão. Portanto, verificaremos primeiro que o algoritmo sempre pára. Digamos que, para calcular o máximo divisor comum entre a e b, fazemos uma seqüência de divisões como abaixo.

$$\begin{aligned} a &= bq_1 + r_1 \text{ e } 0 \leqslant r_1 < b \\ b &= r_1 q_2 + r_2 \text{ e } 0 \leqslant r_2 < r_1 \\ r_1 &= r_2 q_3 + r_3 \text{ e } 0 \leqslant r_3 < r_2 \\ r_2 &= r_3 q_4 + r_4 \text{ e } 0 \leqslant r_4 < r_3 \\ &\vdots \qquad\qquad \vdots \end{aligned}$$

Esqueça por um instante o que está acontecendo na coluna da esquerda. À direita temos uma seqüência de restos, e observamos que *o seguinte é sempre menor que o anterior* mas *todos são maiores ou iguais a zero*. Escrevendo as desigualdades dos restos uma em seguida à outra,

$$b > r_1 > r_2 > r_3 > r_4 \ldots \geqslant 0. \tag{4.1}$$

Como entre b e 0 há apenas uma quantidade finita de inteiros, esta seqüência não pode continuar indefinidamente. Mas ela só chega ao final se um dos restos for zero, e é isto que garante que o algoritmo sempre pára.

Podemos usar o argumento do parágrafo anterior para determinar qual o número máximo de divisões que precisamos fazer para chegar ao máximo divisor comum. Considere novamente a seqüência de restos (4.1). Cada resto é *estritamente menor* que o anterior. Assim, o maior resto possível numa certa divisão é igual ao resto anterior menos 1. Se isto acontecer em cada uma das divisões, teremos que executar b divisões até chegar a um resto igual a zero, e este seria o *pior caso possível*. Logo o número de divisões a serem executadas em cada caso é *no máximo* b.

Na verdade, podemos mostrar que o número máximo de divisões é sempre menor que b (exceto quando $b \leqslant 3$). É preferível formular o problema da seguinte maneira: quais os menores valores (positivos) de a e b para os quais é preciso efetuar n divisões a fim de calcular o $\mathrm{mdc}(a, b)$? Para que a e b sejam os *menores possíveis* é preciso que os quocientes de cada divisão sejam os *menores possíveis*. O menor quociente possível na divisão de dois números

BIBLIOTECA CENTRAL

positivos é 1. Suponhamos que foram efetuadas n divisões para chegar ao resto zero. Assim a seqüência de restos é

$$r_1 > r_2 > r_3 > \ldots r_{n-1} > r_n = 0.$$

Mas já vimos que no pior caso possível o quociente entre quaisquer dois restos consecutivos tem que ser 1. Escrevendo as divisões do fim para o começo obtemos então

$$\begin{aligned} r_{n-2} &= r_{n-1} \cdot 1 \\ r_{n-3} &= r_{n-2} \cdot 1 + r_{n-1} \\ r_{n-4} &= r_{n-3} \cdot 1 + r_{n-2} \\ &\vdots \\ a &= b \cdot 1 + r_1 \end{aligned}$$

Para fixar as idéias, digamos que $r_{n-1} = 1$ e que $n = 10$, a seqüência de restos fornecida pelas equações acima (escrita em ordem decrescente) é então:

$$34,\ 21,\ 13,\ 8,\ 5,\ 3,\ 2,\ 1,\ 1,\ 0.$$

Assim, os menores valores de a e b para os quais temos que efetuar n divisões a fim de calcular $\mathrm{mdc}(a, b)$ (e descobrir que vale 1) são $a = 34$ e $b = 21$. Observe que mesmo b sendo o menor possível neste caso, ainda assim temos $b = 21$ que é muito maior que $n = 10$. A seqüência obtida acima é a famosa *seqüência de Fibonacci*; que aparece também no exercício 5.

5. Demonstração do algoritmo euclidiano

Vimos assim que o algoritmo sempre pára–e executa no máximo tantas divisões quanto o menor dos números de que se quer calcular o máximo divisor comum. Mas por que é que o último resto não nulo coincide com o máximo divisor comum? Para entender isto precisamos de um resultado auxiliar, o que os matemáticos costumam chamar de *lema*. A palavra vem do grego e significa 'alguma coisa que é assumida' a fim de se provar um teorema.

Lema. *Sejam* a *e* b *números inteiros positivos. Suponhamos que existam inteiros* g *e* s *tais que* $a = bg + s$. *Então* $\mathrm{mdc}(a, b) = \mathrm{mdc}(b, s)$.

Precisamos mostrar que a afirmação contida no lema é verdadeira. Antes de fazer isto, porém, usaremos o lema para provar que o último resto não nulo da seqüência de divisões é o máximo divisor comum. Aplicando o algoritmo

a a e b e supondo que o resto nulo ocorre após n divisões, temos

$$\begin{aligned}
a &= bq_1 + r_1 \quad \text{e} \quad 0 \leqslant r_1 < b \\
b &= r_1 q_2 + r_2 \quad \text{e} \quad 0 \leqslant r_2 < r_1 \\
r_1 &= r_2 q_3 + r_3 \quad \text{e} \quad 0 \leqslant r_3 < r_2 \\
r_2 &= r_3 q_4 + r_4 \quad \text{e} \quad 0 \leqslant r_4 < r_3 \\
&\vdots \qquad\qquad \vdots \\
r_{n-4} &= r_{n-3} q_{n-2} + r_{n-2} \quad \text{e} \quad 0 \leqslant r_{n-2} < r_{n-3} \\
r_{n-3} &= r_{n-2} q_{n-1} + r_{n-1} \quad \text{e} \quad 0 \leqslant r_{n-1} < r_{n-2} \\
r_{n-2} &= r_{n-1} q_n \quad \text{e} \quad r_n = 0
\end{aligned} \tag{5.1}$$

Desta vez podemos ignorar a coluna da direita; vamos nos concentrar no que ocorre à esquerda. Da última divisão temos que r_{n-1} divide r_{n-2}. Logo o maior divisor comum entre os dois é o próprio r_{n-1}. Portanto $\text{mdc}(r_{n-2}, r_{n-1}) = r_{n-1}$.

Aqui entra em ação o lema. Aplicando-o à penúltima divisão, concluímos que

$$\text{mdc}(r_{n-3}, r_{n-2}) = \text{mdc}(r_{n-2}, r_{n-1})$$

que, por sua vez, é igual a r_{n-1}. Aplicando novamente o lema, desta vez à ante-penúltima divisão, temos

$$\text{mdc}(r_{n-4}, r_{n-3}) = \text{mdc}(r_{n-3}, r_{n-2})$$

que já vimos que é igual a r_{n-1}. Continuando assim, coluna acima, chegamos finalmente a $\text{mdc}(a, b) = r_{n-1}$; que é o que desejávamos verificar.

Para que não fique nenhum detalhe faltando na nossa prova de que o algoritmo funciona, falta apenas demonstrar o lema. O lema nos diz que assumindo que a, b, g e s estão relacionados por $a = bg + s$ devemos ser capazes de concluir que $\text{mdc}(a, b) = \text{mdc}(b, s)$. Fica mais fácil explicar a demonstração se supusermos que

$$d_1 = \text{mdc}(a, b) \quad \text{e} \quad d_2 = \text{mdc}(b, s).$$

Até aqui não fizemos nada: apenas demos nome aos respectivos máximos divisores comuns. Com esta notação, o que queremos mostrar é que $d_1 = d_2$. Provaremos isto em duas etapas, verificando que $d_1 \leqslant d_2$ e em seguida que $d_2 \leqslant d_1$. Destas duas desigualdades segue a igualdade desejada.

Provaremos que $d_1 \leqslant d_2$; a outra desigualdade é provada de maneira análoga e fica por sua conta. Lembre-se que $d_1 = \text{mdc}(a, b)$. Assim d_1 divide a e d_1 divide b. De acordo com a definição isto significa que existem inteiros u e v tais que

$$a = d_1 u \quad \text{e} \quad b = d_1 v.$$

Vamos substituir estas expressões para a e b na relação $a = bg + s$, que é um dos dados do lema. Obteremos $d_1 u = d_1 v g + s$, ou seja

$$s = d_1 u - d_1 v g = d_1(u - vg).$$

Mas isto significa que d_1 divide s. Recapitulando: como $d_1 = \text{mdc}(a, b)$ temos, em particular, que d_1 divide b. Com um cálculo mostramos que d_1 também divide s. Portanto d_1 é um divisor comum entre b e s. Mas d_2 é o *maior* divisor comum entre b e s. Logo $d_1 \leqslant d_2$, que é o que devíamos provar.

Voltando ao enunciado do lema, observe que precisamos da relação $a = bg + s$, que é análoga à relação entre dividendo, divisor, quociente e resto; mas *não* precisamos saber que s é menor que b. Portanto o fato do resto ser menor que o divisor não é usado para mostrar que o último resto não nulo é o máximo divisor comum. Entretanto precisamos desta condição quando mostramos que o algoritmo pára. Assim as duas condições do *algoritmo de divisão* são necessárias para provar que o *algoritmo euclidiano* funciona.

6. Algoritmo euclidiano estendido

Ainda não extraímos toda a informação de que vamos precisar referente ao máximo divisor comum. Sejam, mais uma vez, a e b inteiros positivos. Digamos que d é o máximo divisor comum entre eles. Então é possível achar inteiros α e β tais que

$$\alpha \cdot a + \beta \cdot b = d. \tag{6.1}$$

Observe que (a não ser em casos triviais) teremos que ou α ou β é um inteiro negativo.

É possível modificar o algoritmo euclidiano de modo a calcular simultaneamente d, α e β, como na relação acima. Este algoritmo é conhecido como *algoritmo euclidiano estendido*. A versão do algoritmo que apresentaremos deve-se a D.E. Knuth, autor de uma série de livros chamada *The art of computer programming*, que é considerada a bíblia da teoria de algoritmos; veja [**Knuth 1981**].

Consideremos o problema de como determinar α e β em (6.1). A primeira impressão é que devemos começar achando d pelo algoritmo euclidiano; tentando depois calcular α e β. Assim poderíamos expressar α e β usando os

valores dos quocientes e restos obtidos nas divisões que efetuamos. É possível por isto em prática, com um pouco de cuidado. Infelizmente este método dá muito trabalho e é muito ineficiente do ponto de vista computacional. O principal problema é que esta estratégia exige que o computador armazene os quocientes e restos de cada divisão. Se os números forem muito grandes, isto pode ser muito oneroso, ou até impossível.

A implementação inventada pelo Knuth é muito mais sutil. A idéia consiste em expressar *cada um* dos restos obtidos na seqüência de divisões em termos de a e b de maneira semelhante à fórmula (6.1), sem esperar chegar ao *último* resto. Isto parece tornar as coisas piores e não melhores. A vantagem é que, para efetuar os cálculos correspondentes a uma certa divisão, basta guardar os dados referentes às duas divisões imediatamente anteriores. Tudo o mais pode ser apagado, deixando assim uma área maior de memória disponível.

Digamos que, calculando o máximo divisor comum entre a e b, obtemos a seqüência de divisões (5.1), que vamos reescrever na forma

$$\begin{aligned}
a &= bq_1 + r_1 \quad \text{e} \quad r_1 = ax_1 + by_1 \\
b &= r_1q_2 + r_2 \quad \text{e} \quad r_2 = ax_2 + by_2 \\
r_1 &= r_2q_3 + r_3 \quad \text{e} \quad r_3 = ax_3 + by_3 \\
r_2 &= r_3q_4 + r_4 \quad \text{e} \quad r_4 = ax_4 + by_4 \\
&\vdots \qquad\qquad \vdots \\
r_{n-3} &= r_{n-2}q_{n-1} + r_{n-1} \quad \text{e} \quad r_{n-1} = ax_{n-1} + by_{n-1} \\
r_{n-2} &= r_{n-1}q_n \quad \text{e} \quad r_n = 0.
\end{aligned} \tag{6.2}$$

Os números $x_1, \ldots, x_{n-1}$ e $y_1, \ldots, y_{n-1}$ são inteiros a determinar. É melhor condensar a informação contida em (6.2) em uma tabela.

restos	quocientes	x	y
a	$*$	x_{-1}	y_{-1}
b	$*$	x_0	y_0
r_1	q_1	x_1	y_1
r_2	q_2	x_2	y_2
r_3	q_3	x_3	y_3
$\vdots$	$\vdots$	$\vdots$	$\vdots$
r_{n-2}	q_{n-2}	x_{n-2}	y_{n-2}
r_{n-1}	q_{n-1}	x_{n-1}	y_{n-1}

A primeira coisa que você vai observar é que introduzimos duas linhas no começo da tabela que 'legalmente' não deviam estar lá. Estas linhas não correspondem a nenhuma divisão. Afinal nem a, nem b são restos. Chamaremos estas linhas de -1 e 0, refletindo assim seu caráter 'fora da lei'. Logo veremos porque são necessárias.

O que queremos, então, fazer? Queremos descobrir como preencher as colunas x e y. Vamos imaginar, por um momento, que já recebemos a tabela preenchida até um determinado ponto: até a $(j-1)$-ésima linha, por exemplo. Começamos a preencher a j-ésima linha dividindo r_{j-2} por r_{j-1} para achar r_j e q_j, de forma que $r_{j-2} = r_{j-1}q_j + r_j$ e $0 \leqslant r_j < r_{j-1}$. Assim

$$r_j = r_{j-2} - r_{j-1}q_j. \tag{6.3}$$

Lendo nas linhas $j-1$ e $j-2$ os valores de x_{j-2}, x_{j-1}, y_{j-2} e y_{j-1}, podemos escrever

$$r_{j-2} = ax_{j-2} + by_{j-2} \quad \text{e} \quad r_{j-1} = ax_{j-1} + by_{j-1}.$$

Substituindo estes valores em (6.3), obtemos

$$\begin{aligned} r_j &= (ax_{j-2} + by_{j-2}) - (ax_{j-1} + by_{j-1})q_j \\ &= a(x_{j-2} - q_jx_{j-1}) + b(y_{j-2} - q_jy_{j-1}). \end{aligned}$$

Portanto

$$x_j = x_{j-2} - q_jx_{j-1} \quad \text{e} \quad y_j = y_{j-2} - q_jy_{j-1}.$$

Observe que, para calcular x_j e y_j, usamos apenas valores contidos nas duas linhas imediatamente anteriores à linha j, além do quociente q_j. Concluindo: sabemos como preencher qualquer linha da tabela, desde que as duas linhas que a precedem sejam conhecidas.

Temos assim um processo recursivo; só nos falta descobrir como lhe dar o empurrão inicial. É aqui que entram em jogo as duas linhas *ilegais* que introduzimos no início da tabela. A razão pela qual estão lá é que é muito fácil calcular os valores de x e y nestes dois casos. Interpretando estas linhas como as demais, deveremos ter

$$a = ax_{-1} + by_{-1} \quad \text{e} \quad b = ax_0 + by_0;$$

que nos sugere escolher

$$x_{-1} = 1, y_{-1} = 0, x_0 = 0 \text{ e } y_0 = 1;$$

e assim podemos dar início à recursão. Tendo executado o algoritmo, e descoberto que o máximo divisor comum corresponde ao resto r_{n-1}, obtemos

$$d = r_{n-1} = ax_{n-1} + by_{n-1}.$$

Ou seja, $\alpha = x_{n-1}$ e $\beta = y_{n-1}$.

Vejamos um exemplo numérico. Tomando $a = 1234$ e $b = 54$ como na seção 4, teremos a seguinte tabela.

resto	quociente	x	y
1234	*	1	0
54	*	0	1
46	22	$1 - 22 \cdot 0 = 1$	$0 - 22 \cdot 1 = -22$
8	1	$0 - 1 \cdot 1 = -1$	$1 - 1 \cdot (-22) = 23$
6	5	$1 - 5 \cdot (-1) = 6$	$-22 - 5 \cdot 23 = -137$
2	1	$-1 - 1 \cdot 6 = -7$	$23 - 1 \cdot (-137) = 160$
0	3	*	*

Portanto $\alpha = -7$ e $\beta = 160$ e

$$(-7) \cdot 1234 + 160 \cdot 54 = 2.$$

Como sempre, precisamos analisar este algoritmo para saber porque pára e porque funciona. Na verdade o *algoritmo euclidiano estendido* é, como o nome indica, uma extensão do algoritmo original. Além da seqüência de divisões, estamos fazendo cáculos adicionais para determinar os valores de x e y a cada passo. Como o algoritmo euclidiano pára, este também pára. Por outro lado, os valores de α e β obtidos ao final são apenas o x e y da última linha da tabela. Logo (6.1) é verificada porque a condição análoga é satisfeita para os valores de x e y correspondentes a cada linha da tabela. O teorema associado é o seguinte.

Teorema. *Sejam* a *e* b *inteiros positivos e seja* d *o máximo divisor comum entre* a *e* b. *Existem inteiros* α *e* β *tais que*

$$\alpha \cdot a + \beta \cdot b = d.$$

Observe que o teorema não diz que os valores de α e β são únicos. Na verdade, existe uma infinidade de pares de inteiros que satisfazem esta equação. Por exemplo, se k é um inteiro qualquer, e $\alpha \cdot a + \beta \cdot b = d$, então

$$(\alpha + kb) \cdot a + (\beta - ka) \cdot b = d,$$

como você pode verificar realizando o cálculo.

Tendo feito todo este esforço para modificar o algoritmo euclidiano de modo a calcular α e β, uma pergunta se impõe: de que serve conhecer estes números? A verdade é que o conteúdo do resto deste livro é a melhor resposta a esta pergunta. Muitos dos principais resultados que estudaremos dependem do conhecimento destes números. Nossa primeira aplicação será à unicidade da fatoração de um inteiro em produto de primos no próximo capítulo. Mas o próprio método de criptografia RSA não seria possível se não existisse uma maneira eficiente de calcular α e β.

7. Exercícios

1. Para cada par a, b de números inteiros abaixo, calcule o máximo divisor comum e determine α, β de modo que $\text{mdc}(a, b) = \alpha \cdot a + \beta \cdot b$.
(1) 14 e 35
(2) 252 e 180
(3) 6643 e 2873
(4) 272828282 e 3242

2. Sendo n um número inteiro maior que 1, verifique as seguintes igualdades:
(1) $\text{mdc}(n, 2n + 1) = 1$
(2) $\text{mdc}(2n + 1, 3n + 1) = 1$
(3) $\text{mdc}(n! + 1, (n + 1)! + 1) = 1$

3. Sejam $n > m$ inteiros positivos. Mostre que se o resto da divisão de n por m é r então o resto da divisão de $2^n - 1$ por $2^m - 1$ é $2^r - 1$.

4. Sejam $n > m$ inteiros positivos. O objetivo desta questão é calcular

$$\text{mdc}(2^{2^n} + 1, 2^{2^m} + 1).$$

(1) Usando que

$$2^{2^{m+1}} - 1 = (2^{2^m} + 1)(2^{2^m} - 1),$$

mostre que $2^{2^n} - 1$ é múltiplo de $2^{2^m} + 1$ quando $n > m$. Qual o quociente desta divisão?
(2) Usando (1), mostre que o resto da divisão de $2^{2^n} + 1$ por $2^{2^m} + 1$ é 2.
(3) Usando (2), determine $\text{mdc}(2^{2^n} + 1, 2^{2^m} + 1)$.

O resultado deste exercício será utilizado no exercício 8 do capítulo 3.

5. Na seqüência de Fibonacci $1, 1, 2, 3, 5, 8, 13 \ldots$, cada número depois do primeiro é soma dos dois anteriores. Denotando por f_n o n-ésimo termo da

seqüência temos portanto que

$$f_n = f_{n-1} + f_{n-2},$$

e que $f_0 = f_1 = 1$.

(1) Mostre que o máximo divisor comum de dois termos consecutivos da seqüência de Fibonacci é sempre 1.

(2) Quantas divisões são necessárias para calcular $\mathrm{mdc}(f_n, f_{n-1})$?

6. Sejam a e b inteiros positivos. O *mínimo múltiplo comum* de a e b é o menor inteiro positivo que é múltiplo de a e que é múltiplo de b. Vamos denotar este número por $\mathrm{mmc}(a, b)$. Mostre que

$$\mathrm{mmc}(a, b) = \frac{ab}{\mathrm{mdc}(a, b)}.$$

7. O objetivo deste exercício é descrever um método para achar uma solução inteira para a equação $ax + by = c$ onde $a, b, c \in \mathbb{Z}$. Isto é, desejamos encontrar inteiros x e y que satisfaçam esta equação, ou determinar que tais inteiros não existem. Seja $d = \mathrm{mdc}(a, b)$. Então existem inteiros a' e b' tais que $a = da'$ e $b = db'$. Logo

$$c = ax + by = d(a'x + b'y).$$

Mostre que a equação só pode ter solução se d dividir c.

Se isto acontecer ponha $c = dc'$ e considere a *equação reduzida* $a'x + b'y = c'$. Mostre que qualquer solução da equação original é solução da reduzida e vice-versa.

Logo para resolver a equação original, basta achar as soluções da equação reduzida. Para isto podemos usar o algoritmo euclidiano estendido para achar α e β inteiros tais que $\alpha \cdot a + \beta \cdot b = 1$. Mostre que, neste caso, a equação reduzida tem soluções $x = c'\alpha$ e $y = c'\beta$.

8. Usando o método descrito no exercício anterior, escreva um programa para determinar uma solução inteira para a equação $ax + by = c$, tendo como entrada os coeficientes a, b e c. A saída do programa deve ser, ou uma solução inteira da equação, ou uma mensagem indicando que tais soluções não existem. Portanto o programa consistirá, essencialmente, de uma implementação do algoritmo euclidiano estendido.

9. Escreva um programa que, tendo como entrada dois inteiros a e b, determina o máximo divisor comum de a e b. Adapte o seu programa para gerar *aleatoriamente* pares de inteiros a e b e calcular o $\mathrm{mdc}(a, b)$. O programa

deve ter como entrada o número total de pares que você deseja testar, e como saída o quociente

$$\frac{\text{total de pares cujo mdc é 1}}{\text{total de pares testados}}.$$

Este quociente dá uma medida da probabilidade de que um par de inteiros escolhido aleatoriamente seja co-primo. Vai ser necessário deixar o programa testar um número muito grande de pares para obter uma boa aproximação da probabilidade. Execute o programa dez vezes para cada valor escolhido para a entrada. Faça uma tabela com esses valores, tendo como entrada 10, 100, 1000, 10000 e 100000 pelo menos. Usando argumentos de teoria da probabilidade é possível mostrar que, testando um número grande de pares, o quociente acima deve ficar próximo de $6/\pi^2$; veja [**Giblin 1993, p. 16**]. Como é que este valor se compara aos resultados experimentais que você obteve?

2

Fatoração única

Dividir para conquistar é uma estratégia muito útil. Por exemplo, qualquer substância pode ser dividida em partes menores, os átomos. Estudando as propriedades de cada tipo de átomo, poderemos desvendar muitas das propriedades das substâncias que deles são constituídas.

Um fenômeno semelhante ocorre com os números inteiros. Os átomos neste caso são os números *primos* e cada inteiro pode ser escrito como um produto de primos. O conhecimento desta decomposição nos levará a descobrir várias propriedades interessantes dos números inteiros. Infelizmente o preço de executar este programa pode ser, na prática, muito alto. Achar a decomposição de um inteiro em primos pode ser um processo muito lento e laborioso.

1. Teorema da fatoração única

Se vamos decompor inteiros em primos, é conveniente começar recordando a definição de número primo. Um número inteiro p é *primo* se $p \neq \pm 1$ e os únicos divisores de p são ± 1 e $\pm p$. Portanto 2, 3, 5 e -7 são primos, mas $45 = 5 \cdot 9$ não é primo. Um número inteiro, diferente de ± 1, que não é primo é chamado de *composto*. Logo 45 é composto.

Observe que a definição de primo exclui os números ± 1. Isto é, os números ± 1 *não são primos*; mas também não são compostos! O que é peculiar a estes números é que são os únicos inteiros que têm inverso multiplicativo. Voltaremos a esta questão ao final da seção. O principal teorema deste capítulo é o seguinte.

Teorema da Fatoração Única. *Dado um inteiro positivo* $n \geqslant 2$ *podemos sempre escrevê-lo, de modo único, na forma*

$$n = p_1^{e_1} \dots p_k^{e_k},$$

onde $1 < p_1 < p_2 < p_3 < \cdots < p_k$ *são números primos e* $e_1, \dots, e_k$ *são*

inteiros positivos.

Este teorema é tão importante que é chamado às vezes de *Teorema Fundamental da Aritmética*. O primeiro a enunciá-lo na forma acima foi C.F. Gauss na §16 de seu famoso livro *Disquisitiones arithmeticæ*. Isto não significa que o fato expresso no teorema não houvesse sido usado implicitamente por matemáticos desde a Grécia Antiga. Hardy e Wright comentam em seu livro de teoria dos números que isto se deve ao fato de Gauss ter sido *'o primeiro a desenvolver a aritmética como uma ciência sistemática'*; veja [**Hardy e Wright 1994, p. 10**].

Os expoentes $e_1, \ldots, e_k$ no teorema são chamados de *multiplicidades*. Assim a multiplicidade de p_1 na fatoração de n é e_1. Em outras palavras, a multiplicidade de p_1 é o *maior* expoente e_1 tal que $p_1^{e_1}$ divide n. Observe também que n tem k fatores primos *distintos*, mas que a quantidade total de fatores primos (distintos ou não) é a soma da multiplicidades $e_1 + \cdots + e_k$.

Note que o teorema nos diz duas coisas. Em primeiro lugar, todo inteiro pode ser escrito como produto de potências de primos. Em segundo lugar, só há uma escolha possível de primos e expoentes para a fatoração de um inteiro dado. Temos portanto que provar duas coisas: que a fatoração *existe* e que é *única*. Provaremos estas duas partes separadamente. Como veremos adiante, é mais fácil provar que a fatoração existe do que provar que é única.

Tendo o enunciado preciso do teorema de fatoração única, fica mais fácil entender porque é necessário excluir ± 1 da definição de número primo. A verdade é que, se não fizéssemos isto não poderíamos falar da unicidade da fatoração no teorema acima. Por exemplo, se 1 fosse primo, então 2 e $1^2 \cdot 2$ seriam duas fatorações *distintas* do número 2. Usando a mesma idéia de multiplicar o número por uma potência de 1 (ou de -1) teríamos uma infinidade de fatorações distintas para cada inteiro. Para excluir este tipo de fatoração trivial, dizemos que ± 1 não são primos.

Finalmente uma questão etimológica, você já se perguntou porque os números primos têm este nome? O nome é uma herança grega e, naturalmente, não se refere a nenhuma relação de parentesco. Os gregos classificavam os números em *primeiros ou indecomponíveis* e *secundários ou compostos*. Os números compostos são secundários por serem formados a partir dos primos. Os romanos apenas traduziram literalmente a palavra grega para primeiro, que em latim é *primus*. É daí que vêm nossos números primos.

2. Existência da fatoração

Vamos mostrar nesta seção que, dado um inteiro $n \geqslant 2$, podemos escrevê-

lo como produto de primos. A melhor maneira de fazer isto é explicitando um algoritmo que, tendo por entrada n, determina seus fatores primos e respectivos expoentes.

Começaremos descrevendo um algoritmo mais simples, para determinar apenas um fator de um inteiro dado. O algoritmo mais ingênuo possível é o seguinte. Tendo n como entrada, tente dividir n por cada um dos inteiros de 2 a $n-1$. Se algum destes inteiros dividir n, então achamos um fator de n. E mais, o menor fator que acharmos desta maneira tem que ser primo.

Vejamos porque esta última afirmação é verdadeira. Seja f um inteiro tal que $2 \leqslant f \leqslant n-1$. Suponhamos que f é o *menor* fator de n e que f' é um fator (maior que 1) de f. Pela definição de divisibilidade, existem inteiros a e b tais que

$$n = f \cdot a \quad \text{e} \quad f = f' \cdot b.$$

Logo $n = f' \cdot ab$. Portanto f' também é fator de n. Como estamos supondo que f é o *menor* fator de n, concluímos que $f \leqslant f'$. Por outro lado, f' é fator de f, o que só pode acontecer se $f' \leqslant f$. Das duas desigualdades segue que $f = f'$. Moral: o único fator de f maior que 1 é o próprio f. Logo f é primo, que é a afirmação original.

Há ainda uma outra observação que precisamos fazer antes de descrever o algoritmo em detalhes. Pelo que vimos o algoritmo consiste em fazer uma busca, começando de 2, para achar um fator de n. Até onde precisamos ir com nossa busca? Que não precisamos passar de $n-1$ é evidente: um número inteiro não pode ter um fator maior que ele próprio. Na verdade, não precisamos procurar fatores maiores que $\sqrt{n}$. Lembre-se que o algoritmo que estamos descrevendo determina o *menor* fator de n maior que 1. Portanto basta verificar que o *menor* fator de n, maior que 1, é sempre menor ou igual a $\sqrt{n}$. Mas cuidado: há um caso em que isto é evidentemente falso. Se n é primo, então seu menor fator maior que 1 é o próprio n. Portanto o que temos que verificar é que se n é composto e se $f > 1$ é seu menor fator então $f \leqslant \sqrt{n}$.

Seja, portanto, n um número composto e $f > 1$ seu *menor* fator. Então, existe um inteiro positivo a tal que $n = fa$. Como f é o menor fator, certamente $f \leqslant a$. Mas $a = n/f$, logo $f \leqslant n/f$. Disto segue que $f^2 \leqslant n$, que é equivalente a $f \leqslant \sqrt{n}$, como queríamos mostrar.

Resumindo o que vimos até agora: o algoritmo deve buscar um número que divida n, começando de 2 e avançando até $\sqrt{n}$. Se n for composto, vamos encontrar o menor fator de n por este método, e este fator é *necessariamente primo*. Se nenhum dos números pesquisados é fator de n, isto significa que n é primo. O algoritmo, na forma do capítulo 1, é o seguinte.

Algoritmo de fatoração.
Entrada: *inteiro positivo* n.
Saída: *inteiro positivo* f *que é o menor fator primo de* n *ou uma mensagem indicando que* n *é primo.*

Etapa 1: *Comece fazendo* $F = 2$.
Etapa 2: *Se* n/F *é inteiro escreva* F *é fator de* n' *e pare; senão vá para a Etapa* 3.
Etapa 3: *Incremente* F *de uma unidade e vá para a Etapa* 4.
Etapa 4: *Se* $F > \sqrt{n}$ *escreva* n *é primo e pare; senão volte à Etapa 2.*

Dado um inteiro $n > 0$, temos assim uma maneira de determinar se n é primo e, se não for, achar um fator de n. Caso n seja primo, já achamos a sua fatoração. Mas se n for composto, queremos achar *todos* os seus fatores primos e respectivas multiplicidades. Para isto basta aplicar o algoritmo acima várias vezes.

Digamos que, aplicando o algoritmo a n achamos o fator q_1. Então q_1 é o menor fator primo de n. A seguir aplicamos o algoritmo ao co-fator n/q_1, determinando assim um segundo fator primo de n, que vamos chamar de q_2. É claro que $q_1 \leqslant q_2$, mas pode acontecer que sejam iguais, basta que q_1^2 divida n. Continuando, aplicamos o algoritmo ao co-fator $n/(q_1q_2)$ e assim por diante. Com isso determinamos uma seqüência crescente de números primos

$$q_1 \leqslant q_2 \leqslant q_3 \leqslant \ldots \leqslant q_s$$

cada um dos quais é fator de n. A esta seqüência corresponde uma outra, formada pelos co-fatores,

$$n > \frac{n}{q_1} > \frac{n}{q_1q_2} > \frac{n}{q_1q_2q_3} > \cdots > 0.$$

Observe que esta última é uma seqüência *estritamente decrescente* de números inteiros positivos. Como há apenas n inteiros entre n e 0, o algoritmo tem que parar depois de no máximo n laços. Na verdade, é fácil verificar que o menor elemento da seqüência de co-fatores é sempre 1. Para poder escrever a fatoração na forma do enunciado do teorema, basta contar quantos fatores primos iguais ocorrem entre os q's. Isto nos permitirá determinar as multiplicidades dos diversos fatores primos.

Vejamos como isto funciona em um exemplo numérico. Digamos que queremos fatorar $n = 450$. Aplicando o algoritmo, descobrimos que o menor fator de 450 é 2. Vamos aplicar o mesmo algoritmo novamente, só que desta

vez ao co-fator de 2 em 450, que é $450/2 = 225$. Fazendo isto descobrimos que 3 é fator de 225 (e portanto de 450). Aplicando o algoritmo a $225/3 = 75$, descobrimos que 3 ainda é fator deste número. Logo 3^2 é fator de 450. Com mais duas aplicações do algoritmo descobrimos que 5^2 é fator de 450 e que tem 1 como co-fator. Logo, fatoramos 450 completamente, obtendo $450 = 2 \cdot 3^2 \cdot 5^2$.

3. Eficiência do algoritmo usual de fatoração

Os algoritmos descritos na seção anterior são simples de entender e programar, mas são muito ineficientes. É fácil ilustrar isto com um cálculo simples relacionado ao *algoritmo de fatoração* descrito na seção 2. O pior caso é aquele em que o algoritmo tem que executar o maior número de laços. Isto ocorre quando n é primo, quando temos que executar $\sqrt{n}$ laços. Para fixar as idéias, vamos considerar um número primo n, de 100 ou mais algarismos. Isto é $n \geqslant 10^{100}$ e, portanto, $\sqrt{n} \geqslant 10^{50}$. Assim, precisaremos executar pelo menos 10^{50} laços para determinar que n é primo pelo algoritmo de fatoração. Para transformar isto em tempo de cálculo, digamos que nosso computador executa 10^{10} divisões por segundo. Este é um número muito alto, que não é atingido no estado atual da tecnologia; mas tudo bem. Com estes dados temos que nosso computador vai precisar de pelo menos $\frac{10^{50}}{10^{10}} = 10^{40}$ segundos para determinar que n é primo. Fazendo as contas, vemos que isto corresponde a 10^{31} anos. Para pôr este número em perspectiva, lembre-se que a idade do Universo (desde o 'Big Bang') é estimada em 20 bilhões de anos; ou seja $2 \cdot 10^{11}$ anos. Podemos concluir, sem receio, que é impossível confirmar que um número de 100 ou mais algarismos é primo usando este algoritmo.

Isto significa que o algoritmo é inútil? Certamente que não. Se vamos fatorar um inteiro sobre o qual nada sabemos, há sempre a possibilidade que tenha um fator primo pequeno, digamos menor que 10^6. O algoritmo acima encontrará um tal fator rapidamente. Na próxima seção veremos um outro algoritmo, que acha rapidamente fatores próximos à raiz quadrada do número a ser fatorado.

Existem muitos outros algoritmos de fatoração, alguns dos quais estudaremos adiante nestas notas. A eficiência destes algoritmos depende do tipo de fator que tem o número que queremos fatorar. É muito importante você entender que *não* existe um algoritmo de fatoração que funcione bem para todos os inteiros: disto depende a segurança do método RSA. Não é inteiramente claro se a inexistência deste algoritmo geral é um problema intrínseco ou tecnológico. Isto é, não sabemos se um tal algoritmo não pode existir, ou se ainda não fomos expertos o suficiente para inventá-lo.

4. Fatoração por Fermat

O algoritmo que vimos na seção anterior é eficiente apenas quando o inteiro n a ser fatorado é divisível por um primo pequeno. Quão 'pequeno' o primo tem que ser depende do computador de que dispomos. Nesta seção vamos descrever um outro algoritmo, que é muito eficiente quando n tem um fator primo que não é muito menor que $\sqrt{n}$. Este algoritmo, inventado por Fermat, é bem menos ingênuo em sua concepção que o da seção 2.

Para começar suporemos que n é ímpar, já que se n for par então 2 é um de seus fatores. A idéia do algoritmo é tentar achar números inteiros positivos x e y tais que $n = x^2 - y^2$. Supondo que encontramos estes números, temos que

$$n = x^2 - y^2 = (x - y)(x + y).$$

Logo $x - y$ e $x + y$ são fatores de n.

Para poder aplicar o algoritmo de Fermat assumiremos que já temos em nosso computador um algoritmo para determinar a raiz quadrada de n. Na verdade, é suficiente obter a parte inteira da raiz quadrada de n; veja a seção 1 do Apêndice para mais detalhes. Se r é um número real, denotaremos sua parte inteira por $[r]$. Por exemplo, $[\sqrt{125}] = 11$ e $[\pi] = 3$. É claro que se r é inteiro então $[r] = r$.

O caso mais fácil do algoritmo de Fermat ocorre quando n é um quadrado perfeito; isto é, quando existe algum inteiro r tal que $n = r^2$. Neste caso temos que r é fator de n. Além disso, na notação acima $x = r$ e $y = 0$. Observe que se $y > 0$ então

$$x = \sqrt{n + y^2} > \sqrt{n}$$

Isto sugere a seguinte estratégia para encontrar x e y.

Algoritmo de Fermat.

Entrada: *inteiro positivo ímpar* n.
Saída: *um fator de* n *ou uma mensagem indicando que* n *é primo.*

Etapa 1: *Comece com* $x = [\sqrt{n}]$*; se* $n = x^2$ *então* x *é fator de* n *e podemos parar.*
Etapa 2: *Caso contrário incremente* x *de uma unidade e calcule* $y = \sqrt{x^2 - n}$.
Etapa 3: *Repita a Etapa 2 até encontrar um valor inteiro para* y*, ou até que* x *seja igual a* $(n + 1)/2$*: no primeiro caso* n *tem fatores* $x + y$ *e* $x - y$*, no segundo* n *é primo.*

O funcionamento do algoritmo ficará mais claro se fizermos um exemplo numérico. Seja $n = 1342127$ o número que queremos fatorar. A variável x é inicializada com a parte inteira da raiz quadrada de n, que neste caso vale $x = 1158$. Mas

$$x^2 = 1158^2 = 1340964 < 1342127,$$

logo passamos a incrementar x de um em um. Fazemos isto até que

$$\sqrt{x^2 - n}$$

seja inteiro, ou x seja igual a $(n+1)/2$, que neste caso vale 671064. É mais fácil resumir isto em uma tabela

x	$\sqrt{x^2-n}$
1159	33,97
1160	58,93
1161	76,11
1162	90,09
1163	102,18
1164	113

Obtivemos assim um inteiro no sexto laço. Portanto $x = 1164$ e $y = 113$ são os valores desejados. Os fatores correspondentes são $x + y = 1277$ e $x - y = 1051$.

5. Demonstração do algoritmo de Fermat

Não é de modo algum claro porque o algoritmo de Fermat funciona, nem porque pára. Observe que é necessário considerar separadamente o que ocorre quando n é composto, e quando n é primo. No primeiro caso precisamos mostrar que existe um inteiro $x > [\sqrt{n}]$ tal que $\sqrt{x^2 - n}$ é um inteiro menor que $(n+1)/2$. Isto significa que se n é composto então o algoritmo pára antes de chegar a $(n+1)/2$. Se n é primo, então é necessário verificar que o único valor de x possível é $(n+1)/2$.

Suponhamos que n pode ser fatorado na forma $n = ab$ onde $a \leqslant b$. Queremos obter inteiros positivos x e y tais que $n = x^2 - y^2$. Em outras palavras

$$n = ab = (x-y)(x+y) = x^2 - y^2.$$

Como $x - y \leqslant x + y$, isto sugere que tomemos $a = x - y$ e $b = x + y$. Resolvendo este sistema de duas equações em duas incógnitas, obtemos

$$x = \frac{a+b}{2} \quad \text{e} \quad y = \frac{b-a}{2}.$$

De fato, expandindo os produtos notáveis verificamos facilmente que

$$\left(\frac{b+a}{2}\right)^2 - \left(\frac{b-a}{2}\right)^2 = ab = n. \tag{5.1}$$

Note que x e y têm que ser números inteiros, mas $(b+a)/2$ e $(b-a)/2$ estão escritos na forma de fração. Porém n é ímpar, por hipótese. Logo a e b, que são fatores de n, têm que ser ímpares. Portanto $b+a$ e $b-a$ são pares e, conseqüentemente, $(b+a)/2$ e $(b-a)/2$ são inteiros. É por isso que precisamos supor que a entrada do algoritmo é sempre um número ímpar.

Se n é primo então só podemos ter $a = 1$ e $b = n$. Com isto $x = (n+1)/2$; e este é o único valor possível para x se n é primo. Resta-nos considerar o caso em que n é composto. Se $a = b$, o algoritmo obtém a resposta desejada já na Etapa 1. Podemos, então, supor que n é composto e não é um quadrado perfeito; isto é, que $1 < a < b < n$. Veremos que, neste caso, o algoritmo vai parar se forem satisfeitas as desigualdades

$$[\sqrt{n}] \leqslant \frac{a+b}{2} < \frac{n+1}{2}, \tag{5.2}$$

que provaremos a seguir.

A desigualdade da direita nos diz que $a+b < n+1$. Substituindo $n = ab$ nesta última desigualdade e subtraindo $b+1$ de ambos os membros, obtemos $a-1 < ab-b$. Já que $a > 1$, podemos ainda cancelar $a-1$ de ambos os membros. Fazendo isto obtemos $1 < b$. Este argumento mostra que $1 < b$ é equivalente à desigualdade original. Como $1 < a < b$ vale por hipótese, provamos que $(a+b)/2 < (n+1)/2$.

Consideremos agora a desigualdade da esquerda. Observe primeiro que, como $[\sqrt{n}] \leqslant \sqrt{n}$, basta verificar que $\sqrt{n} \leqslant (a+b)/2$. Esta desigualdade é verdadeira se, e somente se, $n \leqslant (a+b)^2/4$ é verdadeira. Mas por (5.1)

$$\frac{(b+a)^2}{4} - n = \frac{(b-a)^2}{4},$$

que é sempre um número não negativo. Obtivemos assim que $(a+b)^2/4 - n \geqslant 0$, que é equivalente à desigualdade desejada.

Voltemos ao algoritmo. Lembre-se que a variável x é inicializada com o valor $[\sqrt{n}]$ e que vai sendo incrementada de uma unidade a cada laço. Assim (5.2) nos garante que, se n for composto, chegaremos a $(a+b)/2$ antes de

chegar a $(n+1)/2$. Quando $x = (a+b)/2$,

$$y^2 = \left(\frac{a+b}{2}\right)^2 - n = \left(\frac{b-a}{2}\right)^2$$

pela identidade (5.1). Atingindo este laço, o algoritmo pára, obtendo a e b como fatores. Portanto, se n é composto, o algoritmo sempre pára antes de chegar a $x = (n+1)/2$, tendo determinado fatores de n.

Este algoritmo nos diz uma coisa muito importante sobre o RSA. Lembre-se que a segurança do RSA depende da dificuldade de fatorar a chave pública n, que é igual ao produto de dois primos. A primeira impressão é que basta escolher os primos grandes, para garantir que n é difícil de fatorar. Mas isto *não* é verdade. Por exemplo, se escolhermos os primos grandes, mas próximos, então n é facilmente fatorável pelo algoritmo de Fermat. Voltaremos a esta questão no Capítulo 11.

6. Propriedade fundamental dos primos

Para provar que a fatoração de um inteiro em primos é única precisamos de uma propriedade fundamental dos números primos. Nesta seção vamos verificar esta propriedade; nas seções 7 e 8 veremos algumas de suas aplicações. Começamos com um lema, que é nossa primeira aplicação do teorema decorrente do algoritmo euclidiano estendido.

Lema. *Sejam* a, b *e* c *inteiros positivos e suponhamos que* a *e* b *são primos entre si.*
(1) Se b *divide o produto* ac *então* b *divide* c.
(2) Se a *e* b *dividem* c *então o produto* ab *divide* c.

Considere a primeira afirmação. Temos por hipótese que a e b são primos entre si; isto é, que $\mathrm{mdc}(a,b) = 1$. O algoritmo euclidiano estendido nos garante que existem inteiros α e β tais que

$$\alpha \cdot a + \beta \cdot b = 1.$$

O 'passe de mágica' desta demonstração consiste em multiplicar esta equação por c, obtendo

$$\alpha \cdot ac + \beta \cdot c \cdot b = c. \tag{6.1}$$

É evidente que a segunda parcela da soma é divisível por b. Mas a primeira parcela também é, porque a afirmação (1) inclui a hipótese adicional de que b divide ac. Logo o lado esquerdo de (6.1) é divisível por b. Portanto, c é divisível por b, como queríamos mostrar.

A segunda afirmação pode ser provada a partir da primeira. De fato, se a divide c, podemos escrever $c = at$, para algum inteiro t. Mas b também divide c. Como a e b são primos entre si, segue da afirmação (1) que b tem que dividir t. Assim teremos que $t = bk$, para algum inteiro k. Portanto

$$c = at = a(bk) = (ab)k$$

é divisível por ab, que é a afirmação (2).

Teremos muitas oportunidades de usar as duas partes deste lema, começando pela demonstração de uma propriedade muito importante dos números primos. Já conhecida pelos gregos antigos, esta propriedade aparece como a Proposição 30 do livro VII dos *Elementos* de Euclides.

Propriedade fundamental dos primos. *Seja* p *um número primo e* a *e* b *inteiros positivos. Se* p *divide o produto* ab *então* p *divide* a *ou* p *divide* b.

Para provar isto usaremos o lema. Se por acaso p dividir a, estamos feitos. Digamos, então, que p *não* divide a. Neste ponto usamos o fato de que p é primo, para concluir que se p *não* divide a então p e a são primos entre si. Isto ocorre porque qualquer divisor comum a p e a divide p; mas os únicos divisores positivos de p são 1 e p. Portanto, se p não divide a, então $\text{mdc}(a, p) = 1$. Por isso podemos aplicar o lema: como p e a são primos entre si e como p divide ab temos que p divide b.

7. Números irracionais

A unicidade da fatoração em primos será uma conseqüência da propriedade fundamental da seção anterior. Nesta seção consideraremos uma outra conseqüência; mostraremos que se p é primo, então $\sqrt{p}$ é um número irracional. Esta será a primeira de muitas demonstrações que faremos usando o método conhecido por *redução ao absurdo* ou *demonstração por contradição*. A idéia deste método é muito simples e é aplicada freqüentemente no dia-a-dia. Por exemplo, meu carro vai em alta velocidade e preciso freiá-lo; mas esqueci qual dos dois pedais menores é o freio. Se eu pisar no pedal da esquerda e o carro não parar, é porque o freio corresponde ao pedal da direita...

Vamos ver como isto funciona para provar que $\sqrt{p}$ é irracional. O que é que significa isto? Às vezes se ouve alguém dizer que os irracionais têm este nome porque não podem ser entendidos! Contudo *razão* aqui não significa raciocínio, mas sim

uma relação entre magnitudes similares, com respeito à quantidade, deter-

minada pelo número de vezes que uma contém a outra (seja inteiro ou fracionário).

Esta é uma reprodução mais ou menos fiel da definição 3 do Livro V dos *Elementos* de Euclides. A bem da verdade, trata-se de uma definição um tanto obscura. Felizmente tudo o que precisamos saber é que um número irracional é aquele que *não é uma fração*. Temos assim uma questão típica a que podemos aplicar *redução ao absurdo*. Queremos provar que $\sqrt{p}$ *não é* uma fração? Suponhamos, então, *que é* uma fração e vejamos se isto nos leva a alguma dedução absurda. Se levar, é que nossa hipótese é incorreta: isto é, $\sqrt{p}$ é irracional, afinal de contas. A montagem da demonstração tem que ser cuidadosa. Lembre-se que estaremos supondo (na esperança de chegar a um absurdo) que $\sqrt{p}$ é uma fração. Isto é, que existem inteiros positivos a e b tais que

$$\sqrt{p} = \frac{a}{b}. \tag{7.1}$$

Além disto podemos supor que a fração está reduzida; ou seja, que $\mathrm{mdc}(a, b) = 1$. Toda fração pode ser escrita em forma reduzida: basta cancelar o máximo divisor comum entre o numerador e o denominador. É importante supor que a fração a/b está na forma reduzida porque é a partir daí que chegaremos a um absurdo.

Para trabalhar com inteiros na demonstração, vamos elevar a equação (7.1) ao quadrado. Obtemos

$$p = a^2/b^2$$

isto é

$$b^2 \cdot p = a^2. \tag{7.2}$$

Logo p divide a^2. Pela *propriedade fundamental dos primos* isto implica que p divide a. Em outras palavras, existe um inteiro c tal que $a = pc$. Substituindo em (7.2)

$$b^2 \cdot p = p^2 \cdot c^2.$$

Cancelando p de ambos os membros, concluímos que p divide b^2. Novamente pela propriedade fundamental, temos que p divide b. Mas isto não pode acontecer, já que $\mathrm{mdc}(a, b) = 1$. Chegamos assim ao almejado absurdo, e mostramos que $\sqrt{p}$ é irracional.

A existência dos números irracionais é um problema com uma história muito interessante. Segundo o historiador grego Heródoto, a geometria teve origem no Egito. O ponto de partida teria sido a necessidade de redistibuir terras depois das enchentes do Nilo. Se a matemática se limitasse a medidas

de terrenos e cálculos monetários, poderíamos muito bem supor que todos os números são racionais. Foi somente com o desenvolvimento dos aspectos teóricos da geometria na Grécia, que a existência dos números irracionais foi descoberta.

A descoberta dos irracionais é creditada à escola filosófica (ou seita) fundada por Pitágoras. Os pitagóricos estavam muito empenhados no desenvolvimento teórico da matemática porque, segundo eles, os números (inteiros e suas frações) eram a essência de tudo no universo. Daí se pode imaginar o horror sentido pelos pitagóricos quando um de seus membros descobriu que nem toda medida de comprimento corresponde a uma fração. Hipasus de Metapontum teria sido expulso da seita por ter tornado público este segredo. Não satisfeitos, os pitagóricos lhe erigiram um túmulo enquanto ainda vivia, porque para eles já estava morto.

Inevitavelmente a descoberta dos irracionais logo foi amplamente divulgada, e passou a fazer parte da bagagem de conhecimento dos gregos cultos da época clássica. Por exemplo, Platão em um de seus diálogos (o *Teetetus*) afirma que Teodoro de Cirene havia provado que os números $\sqrt{3}, \ldots, \sqrt{17}$ são irracionais. Infelizmente ele não dá nenhuma indicação de como seria esta demonstração.

A demonstração que vimos acima da irracionalidade de $\sqrt{p}$ tem origem nos matemáticos da Grécia Antiga. Aristóteles discute esta demonstração quando $p = 2$, em um de seus livros. Este caso particular consta também dos *Elementos* de Euclides, como a Proposição 117 do Livro X.

8. Unicidade

É chegada a hora de, sem mais delongas, provar que a fatoração de um inteiro na forma do teorema é única. Nossa demonstração será por absurdo. Suporemos que existe algum inteiro que admite mais de uma fatoração na forma estabelecida pelo teorema, e tentaremos chegar a uma contradição. Vamos chamar de n o *menor* inteiro positivo entre aqueles que têm duas fatorações distintas. Podemos escrever

$$n = p_1^{e_1} \ldots p_k^{e_k} = q_1^{r_1} \ldots q_s^{r_s}. \tag{8.1}$$

onde $p_1 < \cdots < p_k$ e $q_1 < \cdots < q_s$ são primos e $e_1, \ldots, e_k, r_1, \ldots, r_s$ são inteiros positivos. Mas estamos supondo que estas fatorações são diferentes. Isto pode acontecer por duas razões. Os primos da fatoração à direita podem não ser os mesmos da esquerda; ou, se forem os mesmos, podem ter multiplicidades diferentes. Saber qual destas duas hipóteses supostamente

é verificada em (8.1) não é importante para os cálculos que se seguem.

De acordo com a fatoração da esquerda, p_1 é um primo que divide n. Mas $n = q_1^{r_1} \dots q_s^{r_s}$, segundo a fatoração da direita. A *propriedade fundamental dos primos* nos garante então que p_1 deve dividir um dos fatores do produto à direita. Isto significa, em última análise, que p_1 divide um dos primos da fatoração da direita. Mas um primo só pode dividir outro se forem iguais. Logo p_1 tem que ser um dos primos q_1, q_2, ... ou q_k. Para fixar as idéias, digamos que $p_1 = q_j$, onde $1 \leqslant j \leqslant s$.

Com isso podemos reescrever (8.1) substituindo q_j por p_1

$$\begin{aligned} n = p_1^{e_1} \dots p_k^{e_k} &= q_1^{r_1} \dots q_j^{r_j} \dots q_s^{r_s}. \\ &= q_1^{r_1} \dots p_1^{r_j} \dots q_s^{r_s}. \end{aligned}$$

Como temos p_1 em ambos os lados da igualdade com multiplicidade pelo menos 1, podemos cancelá-lo obtendo

$$m = p_1^{e_1-1} \dots p_k^{e_k} = q_1^{r_1} \dots p_1^{r_j-1} \dots q_s^{r_s}. \tag{8.2}$$

que é um novo número, menor que n. Mas m tem duas fatorações, que estão escritas em (8.2). Observe que estas fatorações foram obtidas a partir das fatorações em (8.1)–que são distintas por hipótese–cancelando-se um termo comum–que é p_1. Logo as fatorações de m em (8.2) são necessariamente distintas.

Temos assim um número m menor que n com duas fatorações diferentes. Mas isto não é possível, já que assumimos que n era o *menor* inteiro positivo com duas fatorações distintas. Isto significa que a hipótese de que existem duas fatorações distintas leva a um absurdo, confirmando que a fatoração é única.

Tendo feito todo este trabalho, precisamos confrontar um fato um pouco desagradável. Se paramos para pensar um pouco vemos que a unicidade da fatoração nos parece um tanto ou quanto óbvia. Isto acontece porque estamos acostumados a usá-la sem pensar, como se fosse um reflexo. Tanto é assim que até temos dificuldade de imaginar como seria uma fatoração que não é única.

A verdade é que conjuntos de números para os quais a fatoração existe mas *não* é única são a regra, e não a exceção. Quando falamos de Fermat na introdução, mencionamos o resultado conhecido como *Último Teorema de Fermat*. Este nome refere-se à afirmação de que se $n \geqslant 3$ e x, y e z são inteiros tais que

$$x^n + y^n = z^n \tag{8.3}$$

então $x \cdot y \cdot z = 0$. À margem de sua cópia da *Aritmética* do grego Diofanto, Fermat comentou que tinha uma demonstração deste fato, mas que o espaço no livro era insuficiente para descrevê-la em detalhes. É improvável que a demonstração de Fermat estivesse correta, já que o resultado permaneceu sem demonstração até 1995. Além do mais, os métodos usados por A. Wiles e R. Taylor na sua prova são extremamente sofisticados e não existiam sequer há 50 anos.

Julgando pelos erros cometidos por vários matemáticos na tentativa de provar este teorema, podemos conjecturar como teria sido a 'demonstração' de Fermat. O método consistiria em fatorar a expressão (8.3) usando números complexos. Isto nos tira do âmbito dos inteiros e nos leva a trabalhar em um subconjunto dos números complexos. Todo elemento deste subconjunto pode ser *fatorado em irredutíveis*, de uma maneira muito semelhante à dos inteiros em primos. Se a fatoração fosse única, a demonstração do teorema seria relativamente fácil. Mas a fatoração *não* é única! Pode ter sido aí que Fermat errou. Como alguns matemáticos que vieram depois dele, ele pode ter assumido (implicitamente) que a fatoração num tal subconjunto dos complexos seria única.

Infelizmente, para discutir este assunto com mais cuidado teríamos que fazer uma digressão muito longa (de umas 200 páginas), que nos levaria longe demais das metas destas notas. Se o assunto lhe interessa, consulte o livro [**Edwards 1977**].

9. Exercícios

1. Determine se existem inteiros positivos x, y e z que satisfaçam a equação $2^x \cdot 3^4 \cdot 26^y = 39^z$.

2. Seja $k > 1$ um inteiro. Mostre que os números $k! + 2, k! + 3, \ldots, k! + k$ são todos compostos. Conclua que por maior que seja m sempre existem m números compostos *consecutivos*.

3. Use o algoritmo de Fermat para determinar fatores para os seguintes números: 175557, 455621 e 731021.

4. Diga se cada uma das afirmações abaixo é verdadeira ou falsa. Justifique cuidadosamente suas respostas.

(1) $\sqrt{6}$ é irracional.
(2) A soma de um número irracional com uma fração é sempre irracional.
(3) A soma de dois números irracionais é sempre irracional.

(4) O número $\sqrt{2}+\sqrt{3}$ é racional.

5. Mostre que se n é composto então o número

$$R(n) = \frac{10^n - 1}{9} = \underbrace{111\ldots11}_{n \text{ vezes}}$$

também é composto.

Sugestão: se k é fator de n então $R(k)$ é fator de $R(n)$.

6. Seja $n > 0$ um número inteiro positivo composto e p seu *menor* fator primo. Sabe-se que:

(1) $p \geqslant \sqrt{n}$;
(2) $p - 4$ divide o $\text{mdc}(6n+7, 3n+2)$.

Determine todos os possíveis valores de n.

7. Sejam a e b dois números inteiros positivos. Vamos escrever suas fatorações na forma

$$a = p_1^{e_1} p_2^{e_2} \ldots p_k^{e_k} \text{ e}$$
$$b = p_1^{r_1} p_2^{r_2} \ldots p_k^{r_k},$$

onde $p_1 < p_2 < \cdots < p_k$ são primos e os expoentes $e_1, \ldots, e_k$ e $r_1, \ldots, r_k$ são maiores ou iguais a zero. Observe que *não* estamos afirmando que os mesmos primos aparecem nas fatorações de a e b, já que estamos permitindo que os expoentes sejam nulos. Os únicos primos que podem aparecer nas fatorações de $\text{mdc}(a, b)$ e de $\text{mmc}(a, b)$ são $p_1, \ldots, p_k$; determine as multiplicidades destes primos em cada uma destas fatorações.

8. Dizemos que um número inteiro positivo n é *perfeito* se a soma dos seus fatores (incluindo 1 e n) é $2n$. Por exemplo, 6 e 28 são perfeitos. Suponha que s é um inteiro positivo para o qual o número $2^{s+1} - 1$ é primo.

(1) Mostre que os fatores de $2^s(2^{s+1} - 1)$ se agrupam em duas progressões geométricas de razão 2; uma com primeiro termo 1, a outra com primeiro termo $2^{s+1} - 1$.
(2) Calcule a soma destes fatores e mostre que $2^s(2^{s+1} - 1)$ é perfeito. O resultado acima está contido na proposição 36 do Livro IX dos *Elementos* de Euclides. Por isso os números perfeitos deste tipo são conhecidos como *euclidianos*.

O objetivo das questões 9 e 10 é mostrar que todos os números perfeitos *pares* são euclidianos; isto é, da forma $2^s(2^{s+1}-1)$, onde $2^{s+1}-1$ é primo. Este resultado foi provado pela primeira vez por L. Euler, mas só foi publicado postumamente em 1849. A demonstração que apresentamos pode ser encontrada em [**Dickson 1952, Cap. 1**]. É interessante observar que não se sabe se existem ou não números perfeitos ímpares. Sabe-se, porém, que se um tal número existir, então tem que ser maior que 10^{300} e não pode ter menos de 8 fatores primos.

9. Seja n um número inteiro positivo. Denotaremos por $S(n)$ a soma dos divisores de n, incluindo 1 e n.
(1) Mostre que r é primo se, e somente se, $S(r) = r + 1$.
(2) Mostre que n é perfeito se, e somente se, $S(n) = 2n$.
(3) Sejam b_1 e b_2 inteiros *primos entre si*. Mostre que d é um divisor de b_1b_2 se, e somente se, $d = d_1d_2$ onde $d_1 = \text{mdc}(d, b_1)$ e $d_2 = \text{mdc}(d, b_2)$.
(4) Use (3) para mostrar que se b_1 e b_2 são inteiros positivos *primos entre si* então $S(b_1b_2) = S(b_1)S(b_2)$.

10. Se n é um número perfeito par podemos escrevê-lo na forma $n = 2^st$ onde $s \geqslant 1$ e t é ímpar.
(1) Substitua esta expressão para n em $S(n) = 2n$ e use (4) do exercício anterior para mostrar que 2^{s+1} tem que dividir $S(t)$.
(2) De (1) vimos que $S(t) = 2^{s+1}q$ para algum inteiro positivo q. Mostre que $t = (2^{s+1}-1)q$.
(3) Vamos mostrar *por contradição* que $q = 1$. Suponha que $q > 1$. Então podemos concluir de (2) que t tem pelo menos 3 fatores distintos, que são 1, q e t. Portanto $S(t) \geqslant 1 + q + t$. Mostre que $S(t) = 2^{s+1}q = t + q$ e obtenha uma contradição.
(4) De (3) concluímos que $q = 1$. Substituindo isto nas fórmulas anteriores obtemos $t = 2^{s+1} - 1$ e $S(t) = 2^{s+1}$. Logo $S(t) = t + 1$ e segue de (1) do exercício anterior que t é um número primo.
Agora reúna tudo o que mostramos para concluir que

$$n = 2^s(2^{s+1}-1)$$

onde o segundo fator é primo.

11. Seja n um inteiro positivo e $d(n)$ o número de divisores positivos de n. Dizemos que n é altamente composto se $d(m) < d(n)$ para todo $m < n$. Escreva um programa que, tendo como entrada um número inteiro r, determina todos os números altamente compostos menores que r. Use o pro-

grama para fazer uma lista dos números altamente compostos menores que 5000. Observe as fatorações dos números altamente compostos da sua lista: que propriedades destes números você deduz da observação desta lista? Estes números foram estudados pelo matemático indiano autodidata Srinivasa Ramanujan; veja [**Ramanujan 1927, p. 78**].

12. Escreva um programa que implemente o algoritmo de Fermat para achar dois fatores de um número inteiro positivo que seja menor que 2^{32}. Este exercício é o primeiro de uma seqüência que termina com o exercício 8 do capítulo 11.

3

Números primos

Nos dois primeiros capítulos apenas preparamos o terreno. Estudamos algumas propriedades básicas dos inteiros, sem as quais nada poderíamos provar; e dois algoritmos fundamentais, sem os quais nada poderíamos calcular. Já o assunto deste capítulo está diretamente ligado à nossa principal meta, o entendimento da criptografia RSA. Como vimos, a segurança do RSA depende de sermos capazes de escolher dois números primos grandes–digamos com 60 ou mais algarismos cada um. Neste capítulo discutiremos métodos ingênuos para achar primos. Começaremos com algumas fórmulas para primos, isto é expressões que dão valores primos quando à variável (ou variáveis) é atribuído um valor inteiro positivo. Entre as inúmeras fórmulas existentes, vamos discutir três tipos:

- fórmulas polinomiais;
- fórmulas exponenciais;
- fórmulas fatoriais.

Como conseqüência de nosso estudo das fórmulas fatoriais, provaremos que existem infinitos números primos. Além destas fórmulas, veremos uma outra maneira de gerar primos, o *Crivo de Eratóstenes*. Este é o mais antigo dos métodos para achar primos, e é também o avô de todos os crivos modernos.

1. Fórmulas polinomiais

Um dos tipos mais simples de fórmula para primos que podemos imaginar é a fórmula polinomial. Com isto nos referimos a um polinômio

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$$

cujos coeficientes $a_n, a_{n-1}, \ldots, a_1, a_0$ são números inteiros e que satisfaz a condição

$$f(m) \text{ é primo, para todo inteiro positivo } m.$$

Vamos ver o que acontece no exemplo $f(x) = x^2 + 1$. Tabelando valores inteiros positivos de x contra $f(x)$, temos

x	$f(x)$
1	2
2	5
3	10
4	17
5	26
6	37
7	50
8	65
9	82
10	101

Observe que se x for ímpar, então $f(x)$ é par. Assim, a não ser quando $x = 1$, o valor de $f(x)$ é sempre composto (e múltiplo de 2) quando x é ímpar. Portanto, quando $x > 1$, temos que $f(x)$ só pode vir a ser primo se x for par. Também não adianta considerar apenas valores pares para x, uma vez que $f(8) = 65$ é composto. Vemos, então, que este polinômio não nos dá uma fórmula para primos. Infelizmente este exemplo não é fruto do azar. Pode-se provar o seguinte teorema.

Teorema. *Dado um polinômio* $f(x)$ *com coeficientes inteiros, existe uma infinidade de inteiros positivos* m *tais que* $f(m)$ *é composto.*

Vamos provar o teorema apenas no caso em que o polinômio f tem grau 2. Neste caso ocorrem essencialmente as mesmas dificuldades do caso geral, sem que as idéias sejam obscurecidas pela notação.

Seja $f(x) = ax^2 + bx + c$ um polinômio cujos coeficientes a, b e c são inteiros. Podemos supor que $a > 0$, para que $f(x)$ seja sempre positivo para valores grandes de x. Se $f(x)$ for composto sempre que x for um inteiro positivo, não há nada a provar. Digamos, então, que existe um inteiro positivo m tal que $f(m) = p$ é um número primo (positivo).

Seja agora h um inteiro positivo qualquer. Vamos calcular $f(m + hp)$. Uma pergunta razoável que o leitor atento fará é: de onde veio $m + hp$? A melhor resposta a esta pergunta está no cálculo abaixo. Queremos determinar

$$f(m + hp) = a(m + hp)^2 + b(m + hp) + c.$$

Expandindo o produto notável e pondo p em evidência ficamos com o seguinte

$$f(m+hp) = (am^2 + bm + c) + p(2amh + aph^2 + bh).$$

Observe que a expressão no primeiro parêntesis é igual a $f(m)$, que por sua vez é igual a p, de modo que

$$f(m+hp) = p(1 + 2amh + aph^2 + bh). \tag{1.1}$$

Disto poderíamos apressadamente concluir que $f(m+hp)$ é composto, já que é igual a p vezes um outro inteiro, encerrando assim a demonstração. Infelizmente não é tão simples. Para que $f(m+hp)$ seja composto é preciso que a complicada expressão entre parêntesis à direita de (1.1) não seja igual a 1. Ou seja, queremos saber quais os valores de h para os quais

$$1 + 2amh + aph^2 + bh > 1.$$

Mas esta desigualdade é equivalente a

$$2amh + aph^2 + bh > 0.$$

Como h é positivo *por hipótese*, esta última desigualdade é verificada quando

$$2am + aph + b > 0 \text{ isto é, quando } h > \frac{-b-2am}{ap}.$$

Note que $-b-2am$ pode ser um número positivo, basta que b seja negativo e menor que $-2am$.

O que provamos? Mostramos que se $f(x) = ax^2 + bx + c$ é um polinômio com coeficientes inteiros ($a > 0$) e $f(m) = p$ é primo, então $f(m+hp)$ é composto sempre que $h > (-b-2am)/ap$. Em particular, existem infinitos valores inteiros positivos para x de modo que $f(x)$ seja composto.

A demonstração acima pode ser estendida a um polinômio de qualquer grau. Naturalmente vai ser mais complicado calcular $f(m+hp)$, e precisaremos usar o binômio de Newton. Os cálculos são simples, mas são desagradáveis de escrever por causa dos somatórios. O mais difícil é obter a cota inferior para h. No caso de um polinômio quadrático, acabamos tendo que resolver uma equação linear. Mas se o polinômio for de grau n, vamos nos ver às voltas com um polinômio de grau $n-1$ em h. Você entenderá melhor qual o problema se resolver o exercício 1, que trata do caso $n = 3$. A verdade é que não tentamos obter uma cota explícita no caso geral, mas contornamos o problema usando limites. Para os detalhes, veja [**Ribenboim 1990, Cap. 3, seção II**].

Do que dissemos até agora conclui-se que não existe uma fórmula polinomial para primos, pelo menos se estivermos considerando apenas polinômios em uma variável. Entretanto existem polinômios em mais de uma variável cujos valores positivos são sempre primos. Mas estes polinômios têm grau muito alto e muitas variáveis; consulte [**Ribenboim 1990, Cap. 3, seção III**] para mais detalhes. Infelizmente estas fórmulas são complicadas demais de usar para serem úteis na prática.

2. Fórmulas exponenciais: números de Mersenne

Há duas fórmulas exponenciais de enorme importância histórica. Ambas foram intensamente estudadas pelos matemáticos do século XVII, sobretudo Fermat. As fórmulas são

$$M(n) = 2^n - 1 \quad \text{e} \quad F(n) = 2^{2^n} + 1,$$

onde n é um inteiro não negativo. Os números da forma $M(n)$ são conhecidos como *números de Mersenne*, os da forma $F(n)$ são os *números de Fermat*.

Determinar quais dos números de Mersenne são primos é mais uma questão herdada da Grécia. Na numerologia dos pitagóricos um número é chamado de *perfeito* se é igual à metade da soma de todos os seus divisores positivos. Por exemplo, os divisores de 6 são 1, 2, 3 e 6. Somando estes números obtemos

$$1 + 2 + 3 + 6 = 12 = 2 \cdot 6.$$

Logo 6 é perfeito. Em comparação, nenhum primo é perfeito. De fato, se p é primo, seus divisores são 1 e p e não podemos ter $1 + p = 2p$, a não ser que p seja 1, que não é primo.

Euclides já sabia que os números da forma $2^{n-1}(2^n - 1)$ são perfeitos *quando* $2^n - 1$ *é primo*. Não é difícil provar que todo número perfeito *par* é desta forma. As demonstrações destes resultados podem ser encontradas nos exercícios 8, 9 e 10 do capítulo 2. Assim, para achar os perfeitos pares basta achar os primos de Mersenne. Achar números perfeitos pode parecer um problema um tanto ou quanto irrelevante hoje em dia, mas não esqueça que já tem uns 2500 anos de idade e até hoje não foi satisfatoriamente resolvido. Por exemplo, ninguém sabe se existem ou não números perfeitos ímpares. Sabe-se, porém, que se existirem são enormes e têm muitos divisores primos.

Como vimos na introdução, Marin Mersenne foi um frade e matemático amador do século XVII, que foi correspondente de muitos dos matemáticos famosos da época, entre eles Fermat, Descartes e Pascal. Os números da forma $2^n - 1$ receberam este nome por causa de uma afirmação de Mersenne

que teve grande repercussão nos séculos seguintes. Segundo ele, os números da forma $M(n) = 2^n - 1$ seriam primos quando

$$n = 2, 3, 5, 7, 13, 17, 19, 31, 67, 127 \text{ e } 257;$$

e compostos para os outros 44 valores primos de n menores que 257.

A primeira coisa a observar é que todos os expoentes a que se refere Mersenne são primos. Isto porque se n for composto, então $M(n)$ também será composto. É fácil provar este fato. Se $n = rs$ é composto, então

$$M(n) = 2^n - 1 = 2^{rs} - 1 = (2^r - 1)(2^{r(s-1)} + 2^{r(s-2)} + \cdots + 2^r + 1).$$

Portanto, se r divide n então $M(r)$ divide $M(n)$. A segunda coisa a observar, é que a recíproca é falsa. Em outras palavras, se n for primo isto *não* significa que $M(n)$ tem que ser primo. De acordo com a lista de Mersenne $M(11)$ deve ser composto. De fato,

$$M(11) = 2047 = 23 \cdot 89.$$

Como ocorria freqüentemente na época, Mersenne não apresentou nenhuma justificativa para estes resultados. Em 1732 Euler afirmou que $M(41)$ e $M(47)$ seriam primos. Estes números não fazem parte da lista de Mersenne, só que, neste caso, quem estava errado era Euler! O primeiro erro da lista foi encontrado por Pervusin e Seelhof em 1886. Eles descobriram que $M(61)$ é primo. Outros erros foram encontrados em anos posteriores. Além de $M(61)$, a lista omite os primos $M(89)$ e $M(107)$; e inclui os compostos $M(67)$ e $M(257)$.

No capítulo 9, estudaremos um método, devido a Fermat, que permite encontrar fatores primos para $M(p)$, quando p é primo. Embora este método não seja eficiente quando p é grande, existem muitas outras maneiras de testar se um número de Mersenne é primo. Tecnicamente mais sofisticados, estes outros métodos são também muito mais eficientes. Tanto que alguns dos maiores primos conhecidos são números de Mersenne. Por exemplo, $M(1398269)$ é primo e tem 420921 algarismos.

3. Fórmulas exponenciais: números de Fermat

A história dos números de Fermat é semelhante. Em uma carta de 1640 a um outro matemático amador, o cavalheiro Frenicle, Fermat enumerou os números da forma $F(n) = 2^{2^n} + 1$, para os valores inteiros de n entre 0 e 6. Os números são

$$3, 5, 17, 257, 65537, 4294967297 \text{ e } 18446744073709551617.$$

Em seguida, conjecturou que todos os números desta forma são primos. É muito surpreendente que Fermat não tenha tentado aplicar um método semelhante ao que usou para fatorar números de Mersenne ao último dos números acima, que chamamos de $F(5)$. Foi essencialmente isto que Euler fez, quase cem anos mais tarde, obtendo assim o menor fator primo de $F(5)$. Estudaremos o método de Euler no capítulo 9.

Igualmente curioso é que Frenicle não tenha imediatamente observado o erro de Fermat, já que também ele trabalhara na fatoração de números de Mersenne. A julgar pelo tom da correspondência entre os dois, isto não traria senão satisfação a Frenicle, que aliás estava muito longe de ser um matemático do gabarito de Fermat. Ao contrário, Frenicle parece ter concordado com Fermat quanto a esta conjectura.

Ao passo que os números de Mersenne são uma rica fonte de primos, poucos primos de Fermat são conhecidos. Até hoje não se descobriu nenhum número $F(n)$ primo com $n \geqslant 5$. É claro que calcular com números de Fermat para valores de n 'grandes' é mais difícil do que com números de Mersenne, já que a fórmula que os descreve é duplamente exponencial; isto é, é a exponencial de uma exponencial.

Nestas seções estivemos reduzidos a discutir a história de duas fórmulas exponenciais. Os resultados mais interessantes só serão estudados no capítulo 9, o antepenúltimo. Ao mesmo tempo, encontramos na fórmula de Mersenne uma maneira bem sucedida de produzir primos grandes. Portanto estamos no caminho certo.

A esta altura, já poderíamos desenvolver o método de Fermat, mencionado na seção anterior, para fatorar números de Mersenne. Bastaria que nos limitássemos a usar as técnicas conhecidas no século XVII. Para isto teríamos apenas que provar e pôr em uso algumas identidades. Este é o enfoque adotado, por exemplo, no livro [**Bressoud 1989, Cap. 3**]. Ao invés, adiaremos o estudo deste método até termos desenvolvido as noções básicas de teoria de grupos, que permitirão uma demonstração mais transparente e compacta. Como bônus, veremos que a mesma idéia pode ser aplicada a muitos outros problemas interessantes.

Desenvolver uma teoria geral, da qual problemas específicos são casos particulares é um princípio fundamental em matemática. Desta forma percebemos semelhanças entre resultados que pareciam nada ter em comum. A partir destas semelhanças somos muitas vezes levados a aplicar o mesmo tipo de técnica a problemas que antes pareciam inacessíveis. É por isso que estamos sempre buscando generalizar: para entender melhor e assim enxergar

mais longe.

4. Fórmulas fatoriais

Suponhamos que p é um primo positivo. Construiremos uma função semelhante ao fatorial, só que apenas os primos são multiplicados. Vamos chamá-la de $p^{\#}$. Por definição $p^{\#}$ significará o produto de todos os primos menores ou iguais a p. Por exemplo, $2^{\#} = 2$ e $5^{\#} = 2 \cdot 3 \cdot 5 = 30$. Observe que se $q < p$ são primos *sucessivos* então

$$p^{\#} = q^{\#}p.$$

Estamos interessados nos números da forma $p^{\#}+1$. Para entender porque, observe a tabela abaixo

p	$p^{\#}$	$p^{\#}+1$
2	2	3
3	6	7
5	30	31
7	210	211
11	2310	2311

Todos os números da terceira coluna são primos! Será mera coincidência? Mais ou menos. Paramos maliciosamente em $p = 11$ porque

$$13^{\#} + 1 = 30031 = 59 \cdot 509,$$

é composto.

Embora $p^{\#} + 1$ nem sempre seja primo, podemos mostrar que não tem nenhum fator primo menor do que ou igual a p. Usaremos *redução ao absurdo*. Digamos, então, que $p^{\#} + 1$ é divisível por um primo $q \leqslant p$. Em outras palavras, existe um inteiro positivo r tal que $p^{\#} + 1 = q \cdot r$. É melhor escrever esta equação na forma

$$q \cdot r - p^{\#} = 1. \tag{4.1}$$

Como $q \leqslant p$, então q é necessariamente um fator de $p^{\#}$. Logo q divide ambas as parcelas da diferença em (4.1). Portanto q divide 1; isto é, $q = 1$. Mas isto não é possível porque q é primo.

Concluímos que, mesmo quando $p^{\#} + 1$ é composto, seu menor fator primo tem que ser maior que p. Em princípio isto daria um algoritmo para achar primos. Digamos que conhecemos todos os primos até p. Calculamos então $p^{\#} + 1$ e tentamos fatorá-lo. O problema com este método é que precisamos fatorar $p^{\#}+1$. Mesmo para valores relativamente pequenos de p, este número é muito grande, de modo que fatorá-lo é impraticável.

Se por acaso $p^{\#}+1$ for primo, o problema se torna mais fácil. É geralmente mais econômico determinar se um número é primo do que tentar fatorá-lo. Já comentamos sobre isto na introdução. Veremos no capítulo 6 que existe um método que permite determinar (com um pouco de sorte!) se um número é composto sem tentar achar fatores. Mais adiante, no capítulo 10, veremos como confirmar a primalidade de um número também sem tentar achar fatores.

No caso da fórmula fatorial nada disto é de muita ajuda já que $p^{\#}+1$ raramente é primo. Só são conhecidos 16 primos desta forma, o maior dos quais corresponde a $p = 24029$. Por isso, se entendida como maneira de construir primos, a fórmula fatorial é um desastre.

5. Infinidade dos primos

A verdadeira importância da fórmula fatorial não está na determinação de primos grandes. A razão pela qual a estudamos é que nos permite dar uma demonstração simples do seguinte resultado fundamental.

Teorema. *Existem infinitos números primos.*

A demonstração que veremos aqui consta dos *Elementos* de Euclides como a Proposição 20 do Livro IX. Vamos fazer uma *redução ao absurdo*. Digamos que há apenas uma quantidade finita de primos. Neste caso existe um primo que é o maior de todos; vamos chamá-lo de p. Entretanto, já vimos que o número inteiro $p^{\#}+1$ não pode ter divisores primos $\leqslant p$. Mas estamos supondo que *todos os primos são menores ou iguais a* p. Disto concluímos que o número inteiro $p^{\#}+1$ não tem fatores primos, o que contradiz o *teorema da fatoração única*. Logo existem infinitos números primos.

Existem muitas demonstrações posteriores da infinidade dos primos. Uma das mais importantes é a que foi dada por Euler em 1737, e que esboçamos a seguir. A idéia de Euler consiste em considerar o produto

$$P = \left(\frac{1}{1-1/2}\right)\left(\frac{1}{1-1/3}\right)\left(\frac{1}{1-1/5}\right)\cdots \tag{5.1}$$

onde temos um termo para cada número primo. Se houvesse uma quantidade finita de primos, este produto seria igual a um número real positivo. Assim, procedendo por contradição, é preciso mostrar que isto não é verdade. Euler verificou primeiro que P é igual à soma

$$1+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\frac{1}{5}+\frac{1}{6}+\cdots$$

onde temos, agora, uma parcela para cada inteiro positivo. Não é difícil mostrar que esta soma é divergente, que é o mesmo que dizer que é maior que qualquer número real. Logo, não pode haver apenas uma quantidade finita de primos. Perplexo? Na verdade não é tão difícil assim verificar as afirmações acima, basta um conhecimento básico de análise. Veja, por exemplo, [**Voloch 1987**] ou [**Ingham 1932, Theorem 1, p. 10**].

É claro que se houvesse apenas uma quantidade finita de primos muita coisa seria mais simples, e o mundo seria menos interessante. A existência de infinitos primos sugere muitos problemas importantes. Por exemplo, como se distribuem os primos? À medida que consideramos números cada vez maiores, a 'densidade' dos primos aumenta ou diminui? Há alguma regularidade nesta 'densidade'? uma maneira de medi-la? A melhor maneira de formular este problema é usar a chamada função π. Se x é um número real positivo, definimos $\pi(x)$ como sendo o número de primos positivos menores ou iguais a x. Um problema muito importante na teoria de números é, justamente, o de obter estimativas para $\pi(x)$.

A importância da demonstração de Euler da infinidade dos primos reside no fato de que serviu de inspiração para um artigo fundamental de B. Riemann sobre a distribuição dos primos. Neste artigo, publicado em 1859, Riemann enunciou uma série de resultados relacionados, direta ou indiretamente, à função π; mas deu apenas indicações incompletas sobre as demonstrações. Infelizmente Rieman morreu (de tuberculose) em 1866, sem ter tempo de completar os detalhes de seus resultados. Esta tarefa coube a vários matemáticos, principalmente J. Hadamard.

B. Riemann (1826–1866).

Um dos frutos do esforço de Hadamard para provar os resultados de Rie-

mann foi sua demonstração do famoso *teorema dos números primos*, que afirma que

$$\lim_{x\to\infty} \frac{\pi(x)\log x}{x} = 1,$$

onde $\log x$ é o logaritmo natural de x. Este resultado foi originalmente conjecturado por Gauss no século passado, mas só foi provado, independentemente, por J. Hadamard e C.J. de la Vallé-Poussin em 1896.

Podemos expressar o conteúdo do teorema dos números primos em uma linguagem mais acessível, embora menos precisa, dizendo que se x é muito grande, então $\pi(x)$ deve ter um valor muito próximo de $x/\log x$. Mas para que esta aproximação seja boa é necessário que x seja realmente enorme. Por exemplo, se $x = 10^{16}$, então

$$\pi(x) - \left[\frac{x}{\log x}\right] = 7804289844393,$$

que é da ordem de 10^{13}. Como neste caso $x/\log x$ é da ordem de 10^{14}, temos um erro relativamente grande. A distribuição dos primos é discutida em detalhes nos livros [**Hardy e Wright 1994, Cap. XXII**] e [**Ingham 1932**]. Há muitas outras funções simples que aproximam $\pi(x)$, uma destas é estudada experimentalmente no exercício 11.

6. Crivo de Eratóstenes

O *crivo de Eratóstenes* é o mais antigo dos métodos para achar primos, e não envolve nenhuma fórmula explícita. Como não podia deixar de ser, Eratóstenes foi um matemático grego, e nasceu por volta de 284 a.C.. Apesar de sua proficiência em muitos dos ramos de conhecimento, os contemporâneos de Eratóstenes julgavam que não havia chegado à perfeição em nenhum. Por isso era chamado de 'Beta' (segunda letra no alfabeto grego) e 'Pentatlos'. Dá bem uma medida do gabarito dos matemáticos da Grécia Antiga, que alguém cuja obra sobreviveu 2300 anos fosse conhecido por esses apelidos!

Antes de mais nada, você precisa saber que um *crivo* é uma peneira. Nicômaco em sua *Aritmética*, publicada por volta do ano 100 d.C., introduz o crivo de Eratóstenes da seguinte maneira

O método para obtê-los [os números primos] é chamado por Eratóstenes uma peneira, porque tomamos os números ímpares misturados de maneira indiscriminada e, por este método, como se fosse pelo uso de um instrumento ou peneira, separamos os primos ou indecomponíveis dos secundários ou compostos.

Portanto o *crivo* atua como uma peneira que só deixa passar os números primos. Vejamos como funciona.

Em primeiro lugar o crivo determina todos os primos até um certo inteiro positivo n previamente escolhido. Para realizar o crivo com lápis e papel podemos proceder da seguinte maneira. Listamos os ímpares de 3 a n. É claro que só listamos os ímpares porque 2 é o único primo par. Começamos então a operar com o crivo propriamente dito. O primeiro número da nossa lista é 3; riscamos os demais números da lista, de 3 em 3. Assim serão riscados todos os múltiplos de 3 maiores que ele próprio. Em seguida procuramos o menor elemento da lista, maior que 3, que não tenha sido riscado; que é 5. Riscamos os demais números da lista, de 5 em 5. Assim serão riscados todos os múltiplos de 5 maiores que ele próprio. E assim por diante, até chegar a n.

Por exemplo, se $n = 35$, a lista de números é

$$3 \quad 5 \quad 7 \quad 9 \quad 11 \quad 13 \quad 15 \quad 17 \quad 19 \quad 21 \quad 23 \quad 25 \quad 27 \quad 29 \quad 31 \quad 33 \quad 35$$

Ao final da primeira passagem do crivo (de 3 em 3), ficamos com

$$3 \quad 5 \quad 7 \quad \cancel{9} \quad 11 \quad 13 \quad \cancel{15} \quad 17 \quad 19 \quad \cancel{21} \quad 23 \quad 25 \quad \cancel{27} \quad 29 \quad 31 \quad \cancel{33} \quad 35$$

Ao final da segunda passagem do crivo (de 5 em 5) a lista é

$$3 \quad 5 \quad 7 \quad \cancel{9} \quad 11 \quad 13 \quad \cancel{15} \quad 17 \quad 19 \quad \cancel{21} \quad 23 \quad \cancel{25} \quad \cancel{27} \quad 29 \quad 31 \quad \cancel{33} \quad \cancel{35}$$

Ao final da terceira passagem do crivo (de 7 em 7), a lista continua a mesma acima. A quarta passagem seria de 11 em 11, mas novamente nada vai mudar na lista. Na verdade nenhuma passagem posterior do crivo vai eliminar nenhum número adicional desta lista. Logo os primos ímpares positivos menores que 35 são

$$3 \quad 5 \quad 7 \quad 11 \quad 13 \quad 17 \quad 19 \quad 23 \quad 29 \quad 31.$$

Este exemplo nos leva a observar algumas características do crivo. Em primeiro lugar, alguns números são riscados da lista mais de uma vez. É o caso de 15 que já havia sido riscado na primeira passagem, e foi riscado também na segunda. Em segundo lugar, já havíamos riscado da lista todos os números compostos na segunda passagem do crivo. Todas as passagens posteriores foram redundantes.

Consideremos a segunda observação. Ela indica que deve ser possível parar de riscar os números muito antes de chegar a n. De fato, se m é um inteiro da lista, então $m \leqslant n$. Se m for composto, então terá um fator menor ou

igual a $\sqrt{m}$. Mas $\sqrt{m} \leqslant \sqrt{n}$. Isto é, qualquer número composto da lista tem um fator menor ou igual a $\sqrt{n}$. Desta forma não precisamos riscar números de r em r quando $r > [\sqrt{n}]$. No exemplo acima $[\sqrt{35}] = 6$; por isso é suficiente riscar de 3 em 3 e de 5 em 5, e nada mais.

A outra observação é mais delicada. Infelizmente não é possível evitar completamente o fato de que alguns números serão riscados várias vezes. Mas podemos melhorar um pouco o crivo acima. Digamos que queremos achar os primos até n, e que estamos prestes a riscar os números de p em p para algum primo p. É claro que os múltiplos de p que também são múltiplos de primos menores que p já foram riscados da lista. Portanto, nesta etapa, podemos começar a riscar de p em p a partir do menor múltiplo de p, *que não é múltiplo de um primo menor que* p; isto é, a partir de p^2. Resumindo: podemos riscar de p em p a começar de p^2. Isto evita algumas duplicações e torna o crivo um pouco mais econômico.

Vejamos como programar o crivo. A lista dos ímpares corresponderá a um vetor (ou *array*). Lembre-se que a cada entrada de um vetor estão associados dois números: o valor com que a entrada foi preenchida e o índice que determina sua posição no vetor. Por exemplo, considere o vetor

$$\begin{array}{ccccccccc} (& a & b & c & d & e & f & g &) \\ & & \uparrow & & & & & & \end{array}$$

A entrada indicada pela seta foi preenchida com o valor b, e corresponde à segunda casa do vetor, logo tem índice 2.

Voltemos ao crivo. Digamos que queremos determinar os primos menores ou iguais a n. Começamos criando um vetor de $(n-1)/2$ posições, que é quantos números ímpares existem entre 3 e n. A entrada de índice j do vetor corresponde ao j-ésimo número ímpar positivo, que é $2j+1$. As entradas poderão ter somente dois valores 0 ou 1. Uma entrada estará preenchida com 0 se o número ímpar correspondente ao seu índice foi riscado em alguma passagem anterior do crivo. Portanto, todas as entradas do vetor são inicializadas com 1, já que no início ninguém foi riscado. Riscar $2j+1$ corresponde a substituir o 1 por 0 na posição j do vetor.

Listamos abaixo uma versão do algoritmo correspondente ao crivo. Esta versão inclui os dois melhoramentos mencionados anteriormente. Isto é, risca de p em p começando de p^2 e pára de aplicar o crivo quando $p > \sqrt{n}$.

Crivo de Eratóstenes.
Entrada: *inteiro positivo ímpar* n.
Saída: *lista de primos ímpares* $\leqslant n$.

Etapa 1: *Comece criando um vetor* v *de* $(n-1)/2$ *posições, cada uma das quais deve estar preenchida com o valor* 1*; e fazendo* $P = 3$.
Etapa 2: *Se* $P^2 > n$ *escreva os números* $2j+1$ *para os quais a* j*-ésima entrada do vetor* v *é* 1 *e pare; senão vá para a Etapa 3.*
Etapa 3: *Se a posição* $(P-1)/2$ *do vetor* v *está preenchida com* 0 *incremente* P *de* 2 *e volte à Etapa 2; senão vá para a Etapa 4.*
Etapa 4: *Atribua o valor* P^2 *a uma nova variável* T*: substitua por zero o valor da posição* $(T-1)/2$ *do vetor* v *e incremente* T *de* $2P$*; repita estas duas instruções até que* $T > n$*; quando isto acontecer incremente* P *de* 2 *e volte à Etapa 2.*

Observe que na última linha incrementamos T de $2P$, e não de P, como talvez você esperasse. Lembre-se que só estamos listando os ímpares. Mas se T e P são ímpares, então $T+P$ é par. Logo, quando estamos 'riscando' de P em P, o ímpar a ser riscado em seguida a T é $T+2P$.

Parece haver uma maneira óbvia de melhorar o desempenho deste algoritmo: ao invés de substituir 1 por 0 quando o número é 'riscado', por que não apagá-lo completamente da lista? Isto diminuiria a lista a cada passagem do crivo, o que tornaria o algoritmo mais rápido. Infelizmente não podemos fazer isto. A maneira pela qual sabemos que um determinado número é múltiplo de um primo p depende de sua posição relativa na lista. Isto é, os múltiplos de p aparecem espaçados de p em p. Se apagássemos elementos da lista à medida que o crivo fosse sendo aplicado, isto deixaria de ser verdade

Como todo algoritmo, o *Crivo de Eratóstenes* tem suas limitações. Se o que queremos é achar primos muito grandes, este não é o algoritmo que devemos usar. Mas não podemos creditar isto como um defeito; afinal, o objetivo do algoritmo é achar *todos* os primos até n, e isto não é razoável se n tiver 200 algarismos.

Limitando-se ao que o algoritmo se propõe a fazer, observamos que usa muita memória e executa os laços muitas vezes, o que é ruim. Por outro lado, não é necessário executar nenhuma divisão (que são cálculos lentos), e o algoritmo é muito fácil de programar, o que é muito bom.

7. Exercícios

1. Seja $f(x) = ax^3 + bx^2 + cx + d$ um polinômio de grau 3, onde a, b, c e d são inteiros e $a > 0$. Suponha que existe um inteiro positivo m tal que $f(m) = p$ é um primo (positivo). Determine os valores positivos de h para os quais $f(m + hp)$ é sempre composto.

2. Ache, procedendo por tentativa, a fatoração em primos de $p^{\#} + 1$ quando:

(1) $p = 17$
(2) $p = 13$

Um primo ímpar ou é da forma $4n + 1$ ou da forma $4n + 3$. Isto é, tem resto 1 ou 3 na divisão por 4. Por exemplo, 3, 7, 11 e 19 são da forma $4n + 3$; ao passo que 5 e 13 são da forma $4n + 1$. Os exercícios 3 a 7 têm como objetivo mostrar que existem infinitos números primos da forma $4n + 3$.

3. Mostre que o produto de dois números inteiros da forma $4n + 1$ é da forma $4n + 1$.

4. Verifique que um número primo qualquer diferente de 2 ou é da forma $4n + 1$ ou da forma $4n + 3$.

5. O produto de dois números da forma $4n + 3$ é da forma $4n + 3$?

6. Suponha que $3 < p_1 < \cdots < p_k$ sejam primos da forma $4n + 3$. Usando o exercício 3, verifique que $4(p_1 \ldots p_k) + 3$ tem que ser divisível por um primo da forma $4n + 3$ que não pertence ao conjunto $\{3, p_1, \ldots, p_k\}$.

7. Usando o exercício anterior, mostre que existem infinitos números primos da forma $4n + 3$.

8. Mostramos no exercício 4 do capítulo 1 que se $n > m$ então

$$\text{mdc}(F(n), F(m)) = 1.$$

Isto significa que dois números de Fermat quaisquer não podem ter um fator primo em comum. Use isto para dar uma outra demonstração de que existem infinitos números primos.

9. Mostre que o único inteiro positivo p para o qual p, $p + 2$ e $p + 4$ são primos é 3.

10. O objetivo deste exercício é a elaboração de um programa para determinar alguns valores (inteiros) de x para os quais um polinômio quadrático $f(x)$ assume valores primos. O programa deve ter como entrada os coeficientes a, b e c do polinômio $f(x) = ax^2 + bx + c$. Estes números serão inteiros positivos ou negativos, mas podemos supor que $a > 0$. O programa deve tabelar $f(n)$, para n variando entre 0 e 100, e indicar quais destes valores são primos. Para fazer isto, será necessário executar o crivo de Eratóstenes para achar os primos menores que $|f(0)|$ ou $|f(100)|$–dependendo de qual dos dois for maior. Observe que vai ser necessário limitar os valores de a, b e c, e

verificar se os valores de $f(x)$ não estão ultrapassando a precisão dos inteiros da linguagem de programação que está sendo utilizada. Se você não tomar estes cuidados o programa vai produzir resultados incorretos. Aplique seu programa aos polinômios:

(1) $f(x) = x^2 + 1$,
(2) $f(x) = x^2 - 69x + 1231$,
(3) $f(x) = 2x^2 - 199$,
(4) $f(x) = 8x^2 - 530x + 7681$.

O polinômio (2) é uma variação de um famoso exemplo que L. Euler obteve em 1772.

11. Vimos na seção 5 que existem várias fórmulas que dão aproximações para $\pi(x)$, o número de primos positivos menores ou iguais a x. A fórmula decorrente do teorema dos números primos é $x/\log x$ que não fornece uma boa aproximação a não ser que o valor de x seja enorme. Neste exercício desejamos fazer o estudo experimental de uma outra fórmula que serve de aproximação para $\pi(x)$. A fórmula é:

$$S(x) = \frac{x}{\log x}\left(1 + \left[\sum_{k=0}^{12} a_k(\log\log x)^k\right]^{-1/4}\right)$$

onde log denota o *logaritmo natural* e

$$a_0 = 229168,50747390,\ a_1 = -429449,7206839,\ a_2 = 199330,41355048,$$
$$a_3 = 28226,22049280,\ a_4 = 0,\ a_5 = 0,\ a_6 = -34712,81875914,$$
$$a_7 = 0,\ a_8 = 33820,10886195,\ a_9 = -25379,82656589,$$
$$a_{10} = 8386,14942934,\ a_{11} = -1360,44512548,\ a_{12} = 89,14545378.$$

Escreva um programa, baseado no *Crivo de Eratóstenes* que, tendo como entrada o inteiro x, calcula $\pi(x)$. Use este algoritmo para gerar uma tabela com os valores de $\pi(x) - S(x)$ quando x é igual a 11, 100, 1000, 2000, 3000, ..., 9000 e 10000. Compare com os valores correspondentes de $\pi(x) - x/\log x$. O que você conclui da análise destas tabelas?

12. Já vimos que um primo ímpar ou é da forma $4n + 1$ ou da forma $4n + 3$. Além disto sabemos pelo exercício 7 que há infinitos primos da forma $4n + 3$. Também é verdade que há uma infinidade de primos da forma $4n + 1$, mas a demonstração é mais difícil; veja [**Hardy e Wright 1994, Cap. II, seção**

2.3]. O objetivo deste exercício é estudar experimentalmente a freqüência relativa de cada um destes dois tipos de primos. Seja x um número real positivo. Denotaremos por $\pi_1(x)$ o número de primos positivos da forma $4n+1$ que são menores ou iguais a x; e por $\pi_3(x)$ o número correspondente de primos da forma $4n+3$. Escreva um programa, baseado no *Crivo de Eratóstenes*, que, tendo como entrada um inteiro positivo x, determina $\pi_1(x)$ e $\pi_3(x)$. Tabele $\pi_1(x)$, $\pi_3(x)$ e $\pi_1(x)/\pi_3(x)$ para os valores de x de 100 em 100 entre 0 e 10^5. Sabe-se que $\lim_{x\to\infty} \pi_1(x)/\pi_3(x) = 1$. Os valores da sua tabela condizem com este resultado?

13. Adapte o programa elaborado no exercício anterior para determinar o menor valor inteiro de x para o qual $\pi_1(x) > \pi_3(x)$.

O exame da evidência experimental então disponível levou alguns matemáticos do começo do século à conclusão de que, exceto para alguns valores relativamente pequenos de x, a desigualdade $\pi_1(x) < \pi_3(x)$ seria sempre correta. A verdade emergiu em 1914, quando o matemático inglês J. E. Littlewood mostrou que existem seqüências infinitas

$$x_1, x_2, \ldots \quad \text{e} \quad y_1, y_2, \ldots$$

de números reais positivos, tais que

$$\lim_{i\to\infty} (\pi_1(x_i) - \pi_3(x_i)) = \infty \quad \text{e} \quad \lim_{i\to\infty} (\pi_1(y_i) - \pi_3(y_i)) = -\infty.$$

A moral é óbvia: é sempre muito perigoso generalizar a partir de dados numéricos, mesmo quando a quantidade de dados disponíveis é muito grande.

4

Aritmética modular

Você já ouviu a frase: *esses matemáticos são uns loucos, para eles 2 mais 2 nem sempre é quatro*? Quem fala isso deveria pensar um pouco. Mesmo no dia a dia nem sempre as contas dão como resultado aquilo que reza a aritmética. Por exemplo, quando é que $13 + 18$ dá 7? Quando estamos falando de horas, claro. Se é 1 da tarde, então daqui a 18 horas serão 7 da manhã. Isto não é privilégio das horas; qualquer fenômeno cíclico vai produzir uma aritmética peculiar, semelhante a esta.

Esta aritmética dos fenômenos cíclicos é conhecida como *aritmética modular*. É este o principal tema deste capítulo. Veremos como somar, multiplicar e dividir números nesta nova aritmética; veremos até como resolver equações. Tudo isto porque este é o melhor contexto para estudar as questões de divisibilidade que nos interessam. Vamos começar recordando a noção de *relação de equivalência*.

1. Relações de equivalência

A maneira moderna de introduzir a aritmética modular é através das relações de equivalência. Como estas relações vão desempenhar um papel importante aqui e em outras ocasiões neste livro, é melhor revisar o assunto com cuidado.

Digamos que X é um conjunto; que pode ser finito ou infinito. Definimos uma *relação* em X dizendo como comparar dois elementos deste conjunto. Esta não é exatamente uma definição formal, mas não precisamos mais que isto para os nossos propósitos. Observe que, para definir a relação, precisamos dizer quem é o conjunto subjacente; isto é, o conjunto cujos elementos estão sendo comparados.

Vejamos alguns exemplos. No conjunto dos números inteiros temos duas relações naturais, a relação de igualdade e a relação de desigualdade. Em um conjunto de bolas coloridas, temos a relação *bolas de uma mesma cor*. Aliás este é um exemplo excelente para ter em mente, porque é muito concreto.

As relações de equivalência são de um tipo muito especial. Voltando à situação geral, digamos que X é um conjunto onde está definida uma relação, que denotaremos por $\sim$. Esta é uma *relação de equivalência* se, quaisquer que sejam os elementos $x, y, z \in X$, as seguintes propriedades são satisfeitas:
(1) $x \sim x$.
(2) Se $x \sim y$ então $y \sim x$.
(3) Se $x \sim y$ e $y \sim z$ então $x \sim z$.

A primeira propriedade é talvez a mais estranha. Chama-se *propriedade reflexiva* e nos diz apenas que se a relação é de equivalência então um elemento pode ser comparado a si próprio. É o caso da igualdade dos inteiros. Entretanto a relação $<$ (estritamente menor) no conjunto dos inteiros não satisfaz esta propriedade.

A segunda propriedade chama-se *simétrica*. Novamente a relação $<$ nos inteiros não a satisfaz: $2 < 3$, mas não é verdade que $3 < 2$. Observe ainda que a relação $\leqslant$ (menor ou igual) nos inteiros é reflexiva mas não é simétrica.

A terceira propriedade chama-se *transitiva*. Tanto a igualdade, quanto as relações $<$ e $\leqslant$ nos inteiros são transitivas. Um exemplo de relação que não é transitiva é a relação $\neq$ (diferente) nos inteiros. De fato, temos que $2 \neq 3$ e $3 \neq 2$ mas não é verdade que $2 \neq 2$. Observe que $\neq$ é simétrica mas não é reflexiva.

Insistimos em dar exemplos onde alguma destas propriedades é falsa porque esta é a única maneira de entender o que realmente significam. Exemplos de relações de equivalência não faltam: igualdade nos inteiros, a relação 'bolas de uma mesma cor' em um conjunto de bolas coloridas, a relação 'mesmo número de lados' no conjunto dos polígonos, a relação 'mesmo volume' no conjunto dos sólidos geométricos e assim por diante.

As relações de equivalência são usadas para classificar os elementos de um conjunto em subconjuntos com propriedades semelhantes. E se há uma mania humana, é a mania de classificar tudo. As subdivisões de um conjunto produzidas por uma relação de equivalência são conhecidas como *classes de equivalência*. É conveniente sermos mais formais. Seja X um conjunto e $\sim$ uma relação de equivalência definida em X. Se $x \in X$ então a *classe de equivalência* de x é o conjunto dos elementos de X que são equivalentes a x por $\sim$. Denotando por $\overline{x}$ a classe de equivalência de x temos em símbolos

$$\overline{x} = \{y \in X : y \sim x\}.$$

Por exemplo, escolha uma bola vermelha em um conjunto $\mathcal{B}$ de bolas coloridas. A classe de equivalência desta bola pela relação 'bolas de uma mesma cor' é

o subconjunto das bolas vermelhas contidas no conjunto $\mathcal{B}$.

Há uma propriedade das classes de equivalência que é tão importante, que vamos enunciá-la como um princípio: *qualquer elemento de uma classe de equivalência é um representante de toda a classe*. Isto é, se conhecemos um elemento da classe podemos reconstruir a classe inteira. Imagine que, no exemplo das bolas coloridas, alguém nos diz que o subconjunto que temos diante de nós é uma classe de equivalência. Para saber que classe é esta basta tomar uma bola e ver qual a sua cor, isto é, com um elemento só identificamos a classe inteira.

Voltemos ao conjunto X com a relação de equivalência $\sim$. Para o conjunto X, o princípio acima, nos diz que se y é um elemento da classe de x então as classes de x e y são iguais. Em símbolos,

$$\text{se } x \in X \text{ e } y \in \overline{x} \text{ então } \overline{x} = \overline{y}.$$

Podemos provar isto usando as propriedades das relações de equivalência. Se $y \in \overline{x}$, então, por definição, $y \sim x$; e pela simétrica, $x \sim y$. Mas se $z \in \overline{x}$ então também temos $z \sim x$. Logo pela propriedade transitiva $z \sim y$. Portanto $z \in \overline{y}$. Mostramos assim que $\overline{x} \subseteq \overline{y}$. Um argumento análogo mostra que $\overline{y} \subseteq \overline{x}$. Pedante? Talvez, mas isto é tão importante e tão mal entendido que vale a pena fazer todos estes detalhes. Aliás, falando em pedante, você já se deu conta de que a propriedade reflexiva garante que x é um elemento de $\overline{x}$, sua classe de equivalência?

Para encerrar esta preleção, considere as seguintes propriedades do conjunto X com a relação de equivalência $\sim$:

(1) X é a união de todas as classes de equivalência.
(2) Duas classes de equivalência distintas não podem ter um elemento em comum.

A primeira propriedade é uma conseqüência do fato de que cada elemento pertence à sua própria classe de equivalência. A segunda propriedade segue do princípio enunciado acima.

O conjunto das classes de equivalência de $\sim$ em X tem um nome especial, é chamado de *conjunto quociente* de X por $\sim$. Observe que os elementos do conjunto quociente são subconjuntos de X–as classes de equivalência. Isto é, o conjunto quociente não é um subconjunto de X, mas sim do conjunto das partes de X. Lembre-se que o conjunto das partes de X é aquele cujos elementos são todos os subconjuntos de X.

Vamos encerrar esta revisão com um outro exemplo; nele revelamos a verdadeira natureza deste objeto tão mal entendido, as frações. De que

substância é feita uma fração? De dois números inteiros, é claro; um par de inteiros, no jargão matemático. É claro que interpretamos este par como um quociente, *quando isto é necessário*. A verdade, às vezes cuidadosamente escondida, é que calculamos com frações formalmente, como pares de números, sem recorrer a esta interpretação.

Frações: pares de números? Nada disto. Os pares $(2,4)$ e $(1,2)$ não são iguais, mas representam as mesmas frações. Logo não é tão simples assim. Está preparado para uma revelação? Uma fração é na verdade uma classe de equivalência de pares de números inteiros.

Vamos ser mais cuidadosos. Considere o conjunto $\mathcal{X}$ dos pares (a, b) onde a e b são inteiros e $b \neq 0$. Definimos neste conjunto uma relação de equivalência da seguinte maneira. Os pares (a, b) e (a', b') são equivalentes se $ab' = a'b$. Fica por sua conta verificar que isto define uma relação de equivalência no conjunto $\mathcal{X}$. Uma fração é uma classe de equivalência neste conjunto. A razão pela qual podemos pensar em uma fração olhando somente para um representante a/b (que é o par (a, b)) é que *qualquer representante de uma classe de equivalência nos permite reconstruir toda a classe*–o princípio enunciado anteriormente. Assim o conjunto dos números racionais é, quem diria, o conjunto quociente de $\mathcal{X}$ pela relação definida acima.

Esta maneira de construir frações pode lhe parecer pateticamente complicada, mas tem muitas vantagens. Além disso, nossa versão da aritmética modular vai conduzir a uma situação semelhante: falaremos dos representantes individuais, mas estaremos sempre pensando nas classes.

2. Inteiros módulo n

Vamos considerar a afirmação: *'um dia tem 24 horas'*, do ponto de vista matemático. Ora, um dia de 24 horas é uma subdivisão arbitrária e imaginária de um contínuo de tempo. Podemos descrever isto matematicamente do seguinte modo. Fixamos um momento inicial a partir do qual o tempo é contado. A seguir estabelecemos uma relação de equivalência: dois momentos que diferem por 24 horas correspondem a horas análogas em dias diferentes. Embora o seu verdureiro possa não concordar, esta é uma interpretação muito razoável do que está acontecendo nesta situação.

Queremos fazer uma coisa semelhante, não com a sucessão das horas, mas sim com o conjunto dos números inteiros. O conjunto dos inteiros já vem com um marco inicial natural, o número zero. Vamos escolher um número inteiro positivo, que estará fixado a partir deste momento. Para que não precisemos nos comprometer nesta escolha, chamemos de n este número. O número n

será o *módulo* ou *período* da nossa construção.

Vamos agora construir uma relação de equivalência no conjunto dos inteiros. Diremos que, pulando de n em n, todos os inteiros são equivalentes; ou ainda, dois inteiros cuja diferença é um múltiplo de n são equivalentes. Formalmente, diremos que dois inteiros a e b são *congruentes módulo* n se $a - b$ é um múltiplo de n. Se a e b são conguentes módulo n, escrevemos

$$a \equiv b \pmod{n}.$$

Alguns exemplos numéricos ajudarão a compreender melhor o que está em jogo. Vamos escolher $n = 5$ como módulo, então

$$10 \equiv 0 \pmod{5} \quad \text{e} \quad 14 \equiv 24 \pmod{5}.$$

Outro exemplo, com outro módulo; digamos que $n = 7$, então

$$10 \equiv 3 \pmod{7} \quad \text{e} \quad 14 \equiv 0 \pmod{7}.$$

Observe que quem é congruente a quem depende do módulo que foi escolhido.

Não podemos deixar de verificar que a congruência módulo n é uma relação de equivalência. Começamos com a propriedade reflexiva. Seja a um inteiro. Para mostrar que $a \equiv a \pmod{n}$, temos que verificar, por definição, que a diferença $a - a$ é um múltiplo de n. Mas isto é claro, 0 é múltiplo de qualquer inteiro. A propriedade simétrica também é fácil. Se $a \equiv b \pmod{n}$, então $a - b$ é um múltiplo de n. Mas $b - a = -(a - b)$; logo $b - a$ também é múltiplo de n. Portanto $b \equiv a \pmod{n}$.

Quanto à propriedade transitiva, suponhamos que $a \equiv b \pmod{n}$ e $b \equiv c \pmod{n}$; onde a, b e c são inteiros. A primeira congruência nos diz que $a - b$ é múltiplo de n; a segunda que $b - c$ é múltiplo de n. Somando múltiplos de n temos de volta múltiplos de n; logo $(a - b) + (b - c) = (a - c)$ é um múltiplo de n. Portanto $a \equiv c \pmod{n}$. Assim verificamos as três propriedades e podemos concluir que a congruência módulo n é uma relação de equivalência.

O conjunto que de fato nos interessa é o conjunto quociente de $\mathbb{Z}$ pela relação de congruência módulo n. Este conjunto tem uma notação própria, $\mathbb{Z}_n$; e um nome especial, *conjunto dos inteiros módulo* n. Precisamos identificar os elementos de $\mathbb{Z}_n$. Sabemos, por definição que são subconjuntos de $\mathbb{Z}$: as classes de equivalência da congruência módulo n. Seja $a \in \mathbb{Z}$. A classe de a é formada pelos $b \in \mathbb{Z}$ que satisfazem $b - a$ é múltiplo de n; isto é $b - a = kn$, para algum $k \in \mathbb{Z}$. Podemos assim descrever a classe de a na forma

$$\overline{a} = \{a + kn : k \in \mathbb{Z}\}.$$

Em particular $\overline{0}$ é o conjunto dos múltiplos de n.

Isto produz uma situação curiosa. Se $a \in \mathbb{Z}$, então podemos dividi-lo por n, obtendo q e r inteiros tais que

$$a = nq + r \quad \text{e} \quad 0 \leqslant r \leqslant n - 1.$$

Logo $a - r = nq$ é um múltiplo de n. Portanto $a \equiv r \pmod{n}$. Isto é, um inteiro qualquer é congruente módulo n a um inteiro no intervalo que vai de 0 a $n - 1$. Em outras palavras, o conjunto quociente $\mathbb{Z}_n$ é formado pelas classes $\overline{0}, \overline{1}, \ldots, \overline{n-1}$. Além disso, duas destas classes não podem ser iguais: a única maneira de dois números entre 0 e $n - 1$ serem congruentes módulo n é se forem iguais. Resumindo

$$\mathbb{Z}_n = \{\overline{0}, \overline{1}, \ldots \overline{n-1}\}.$$

Quando uma classe de $\mathbb{Z}_n$ estiver representada na forma $\overline{a}$ com $0 \leqslant a \leqslant n - 1$, diremos que está na *forma reduzida*.

Como devemos imaginar $\mathbb{Z}_n$? Geralmente pensamos em $\mathbb{Z}$ como sendo o conjunto dos pontos marcados ao longo de uma reta horizontal, de uma em uma unidade. Imagine agora que enrolamos esta reta em uma circunferência, colando o ponto n ao ponto 0. Como a reta é infinita, continuamos a enrolá-la na circunferência. Desta maneira os pontos cujas coordenadas são múltiplos de n coincidem todos com o ponto zero. A imagem geométrica correspondente a $\mathbb{Z}_n$ é, portanto, a de uma circunferência, onde estão marcados n pontos eqüidistantes. Cada ponto corresponde a uma das classes de equivalência de $\mathbb{Z}_n$.

3. Aritmética modular

Esta imagem geométrica é muito útil quando se trata de definir a soma de elementos de $\mathbb{Z}_n$. Vamos imaginar que temos um 'relógio de n horas, com um ponteiro'. Isto é, imagine as classes $\overline{0}$ a $\overline{n-1}$ dispostas ao longo de uma circunferência, a intervalos iguais, no sentido horário; e vamos dotar este relógio de um ponteiro, fixado no centro da circunferência. Desta forma construímos uma *máquina de calcular mecânica* para operar em $\mathbb{Z}_n$.

Vejamos como funciona a máquina. Para fazer isto temos que regredir um pouco no tempo e lembrar como calculávamos somas no começo do primário. Para somar 4 e 3 eu fazia o seguinte. Contava quatro dedos na mão esquerda, do mínimo ao indicador. Depois contava mais três, começando com o polegar da mão esquerda e logo passando à direita: polegar e indicador. Agora é só contar quantos dedos temos no todo. É difícil de explicar, mas fácil de fazer;

pelo menos quando os números são pequenos.

Pois bem, é usando a idéia do nostálgico parágrafo acima que vamos somar em $\mathbb{Z}_n$ usando nosso 'relógio'. Por exemplo, para somar duas classes $\overline{a}$ e $\overline{b}$, colocamos o ponteiro na classe a e depois movemos o ponteiro de b unidades no sentido horário. O resultado da soma é a classe para a qual o ponteiro está apontando depois desta operação. Por exemplo, para somar $\overline{5}$ e $\overline{4}$ em $\mathbb{Z}_8$, colocamos o ponteiro em $\overline{5}$ e o movemos, no sentido horário, quatro casas. Movendo três casas estamos de volta em $\overline{0}$, mais uma casa e o ponteiro aponta para $\overline{1}$. Logo $\overline{4} + \overline{5} = \overline{1}$ em $\mathbb{Z}_8$.

Naturalmente não queremos ficar atados a uma determinada máquina de calcular na hora de executar nossas somas. Ainda mais porque uma máquina deste tipo não é praticável quando n é grande. Precisamos descrever esta mesma operação de modo matemático. Isto é fácil. Sejam $\overline{a}$ e $\overline{b}$ as classes de $\mathbb{Z}_n$ que desejamos somar. A fórmula para a operação é a seguinte

$$\overline{a} + \overline{b} = \overline{a + b}.$$

É preciso interpretar esta fórmula corretamente. À esquerda temos a soma de duas classes; à direita temos a classe da soma de dois números inteiros. Assim definimos a operação soma de classes usando uma operação que já é nossa conhecida, a soma de inteiros. Voltemos ao exemplo em $\mathbb{Z}_8$. De acordo com esta fórmula, para somar $\overline{5}$ a $\overline{4}$, somamos os inteiros 4 e 5, obtendo o resultado 9; logo $\overline{5} + \overline{4} = \overline{9}$. O resultado deu diferente da soma usando o relógio? Não, não deu. Como $9 - 1 = 8$ temos que 9 e 1 estão na mesma classe de equivalência módulo 8; isto é $\overline{9} = \overline{1}$.

Esta última observação aponta para um problema importante. Estamos definindo a soma de classes, que são subconjuntos de $\mathbb{Z}$, mas a soma foi definida usando um elemento de cada subconjunto. Quem nos garante que escolhendo elementos distintos de cada classe não obtemos um resultado diferente? Este é um ponto relativamente sutil, mas é essencial compreender o que está acontecendo; do contrário quase tudo até o fim desta seção vai parecer total *nonsense* a você. Vamos voltar ao exemplo de $\mathbb{Z}_8$. Somamos as classes $\overline{5}$ e $\overline{4}$ usando o elementos 5 da primeira e 4 da segunda. Mas $\overline{13} = \overline{5}$ e $\overline{12} = \overline{4}$. O que aconteceria se somássemos as classes $\overline{13}$ a $\overline{12}$? Se estamos mesmo somando as classes o resultado *tem que dar* $\overline{1}$. De acordo com a fórmula temos

$$\overline{13} + \overline{12} = \overline{25}.$$

Como $25 - 1 = 24$ é divisível por 8, então $\overline{25} = \overline{1}$ e obtivemos o mesmo resultado.

Uma maneira de não ter que se preocupar com este problema é exigir sempre que as classes estejam reduzidas na hora de fazer as contas. Mas isto é deselegante e desconfortável: já imaginou se a soma de frações só pudesse ser feita com as frações na forma reduzida? Felizmente não precisamos nos preocupar com isto: *qualquer que sejam os representantes escolhidos para efetuar a soma de duas classes, o resultado sempre é a mesma classe*. Observe: mesma classe mas não necessariamente mesmo representante. Isto é fácil de verificar. Digamos que em $\mathbb{Z}_n$ temos duas classes $\overline{a}$ e $\overline{b}$. Digamos ainda que $\overline{a} = \overline{a'}$ e $\overline{b} = \overline{b'}$. Queremos verificar que $\overline{a+b} = \overline{a'+b'}$. Mas $\overline{a} = \overline{a'}$ é equivalente a dizer que $a - a'$ é múltiplo de n; e o mesmo vale para $b - b'$. Somando dois múltiplos de n temos um múltiplo de n, logo

$$(a - a') + (b - b') = (a + b) - (a' + b')$$

é múltiplo de n. Portanto, $\overline{a+b} = \overline{a'+b'}$, como queríamos mostrar.

A *diferença* entre duas classes é definida de modo análogo e não oferece maior dificuldade. Vamos passar à multiplicação. Neste caso não há uma definição mecânica, como o 'relógio' para a soma. Mas podemos copiar a definição matemática usada para a soma. A fórmula para a multiplicação das classes $\overline{a}$ e $\overline{b}$ de $\mathbb{Z}_n$ é,

$$\overline{a} \cdot \overline{b} = \overline{ab}.$$

Como no caso da soma, temos que verificar que esta fórmula dá um resultado que é indepedendente da escolha de representantes para as classes. Digamos que $\overline{a} = \overline{a'}$ e que $\overline{b} = \overline{b'}$. Queremos mostrar que $\overline{ab} = \overline{a'b'}$. Como $\overline{a} = \overline{a'}$, temos que $a - a'$ é um múltiplo de n; digamos que $a = a' + rn$, para algum inteiro r. De maneira análoga, $b = b' + sn$, para algum inteiro s. Multiplicando,

$$ab = (a' + rn)(b' + sn) = a'b' + (a's + rb' + srn)n.$$

Logo $ab - a'b'$ é um múltiplo de n. Portanto $\overline{ab} = \overline{a'b'}$, que é o que queríamos provar.

Sabemos assim somar e multiplicar classes de $\mathbb{Z}_n$. Essas operações têm propriedades muito semelhantes às operações correspondentes em $\mathbb{Z}$. Sejam $\overline{a}$, $\overline{b}$ e $\overline{c}$ elementos de $\mathbb{Z}_n$. Temos as seguintes propriedades para a adição:

$$(\overline{a} + \overline{b}) + \overline{c} = \overline{a} + (\overline{b} + \overline{c})$$

$$\overline{a} + \overline{b} = \overline{b} + \overline{a}$$

$$\overline{a} + \overline{0} = \overline{a}$$

$$\overline{a} + \overline{-a} = \overline{0}$$

e as seguintes para a multiplicação:

$$(\overline{a} \cdot \overline{b}) \cdot \overline{c} = \overline{a} \cdot (\overline{b} \cdot \overline{c})$$
$$\overline{a} \cdot \overline{b} = \overline{b} \cdot \overline{a}$$
$$\overline{a} \cdot \overline{1} = \overline{a}$$

O elemento $\overline{-a}$ é chamado o *simétrico* de $\overline{a}$ para a operação de soma, exatamente como no caso dos inteiros. Supondo que $\overline{a}$ está na forma reduzida, a forma reduzida de $\overline{-a}$ é $\overline{n-a}$. Observe que as propriedades da adição e multiplicação se correspondem, com exceção da propriedade referente ao simétrico, que não tem análogo para a multiplicação. Voltaremos a isto mais tarde.

Há ainda uma propriedade que relaciona as duas operações, a *distributividade*:

$$\overline{a} \cdot (\overline{b} + \overline{c}) = \overline{a} \cdot \overline{b} + \overline{a} \cdot \overline{c}.$$

A demonstração dessas propriedades segue imediatamente das propriedades correspondentes para as operações com inteiros, e não vale a pena fazê-las.

A esta altura você vai estar pensando: 'tudo bem, exceto por uma pequena diferença nas definições, as operações em $\mathbb{Z}_n$ são como as operações em $\mathbb{Z}$. Até as propriedades são iguais. Não há muito o que se preocupar'. Aqui vai o exemplo que dá cabo a este otimismo. É um exemplo em $\mathbb{Z}_6$. As classes $\overline{2}$ e $\overline{3}$ são, evidentemente, diferentes da classe $\overline{0}$. Entretanto

$$\overline{2} \cdot \overline{3} = \overline{6} = \overline{0}.$$

O produto de duas classes não nulas pode ser a classe $\overline{0}$! Isto é sinal de que $\mathbb{Z}_n$ tem uma aritmética muito mais complicada do que parecia; mais complicada e mais interessante. Voltaremos a falar sobre divisão em $\mathbb{Z}_n$ em breve, antes vejamos algumas aplicações destas idéias.

4. Critérios de divisibilidade

Verificar que um número é divisível por 3 todo mundo sabe fazer: basta somar os algarismos do número. Se a soma é divisível por 3 então o número original também é. Mas porque isto é verdade? Podemos provar isto usando congruências módulo 3.

Seja a um número inteiro. Queremos verificar se a é divisível por 3. Precisamos conhecer os algarismos de a. Digamos que

$$a = a_n a_{n-1} \ldots a_1 a_0,$$

onde a_0 é o algarismo das unidades, a_1 o algarismo das dezenas, e assim por diante. Em outras palavras

$$a = a_n 10^n + a_{n-1} 10^{n-1} + \cdots + a_1 10 + a_0.$$

Observe agora, que $10 \equiv 1 \pmod 3$. Logo qualquer potência de 10 é congruente a 1 módulo 3. Portanto,

$$a \equiv a_n + a_{n-1} + \cdots + a_1 + a_0 \pmod 3.$$

Se o segundo termo da congruência acima é divisível por 3, então

$$a_n + a_{n-1} + \cdots + a_1 + a_0 \equiv 0 \pmod 3.$$

O que nos permite concluir que $a \equiv 0 \pmod 3$; isto é, a é divisível por 3. Verificamos assim que um número é divisível por 3 se, e somente se, a soma dos seus algarismos também é.

Observe que todos os cálculos acima continuam sendo verdade se substituirmos 3 por 9, já que $10 \equiv 1 \pmod 9$. Portanto, um número inteiro é divisível por 9 se a soma dos seus algarismos é divisível por 9. Podemos usar o mesmo tipo de argumento para provar outros critérios de divisibilidade. Vejamos o que obtemos para 11.

Mais uma vez

$$a = a_n 10^n + a_{n-1} 10^{n-1} + \cdots + a_1 10 + a_0,$$

onde

$$a_n, a_{n-1}, \ldots, a_1, a_0$$

são os algarismos de a. Observe que $10 \equiv -1 \pmod{11}$, portanto

$$10^k \equiv (-1)^k \pmod{11}$$

é igual a 1 ou -1 dependendo da paridade de k. Assim

$$a \equiv a_n(-1)^n + a_{n-1}(-1)^{n-1} + \cdots + a_2 - a_1 + a_0 \pmod{11}.$$

Ao invés da soma dos algarismos, obtivemos a soma *alternada* dos algarismos. Portanto, um número é divisível por 11 se, e somente se, a soma alternada dos seus algarismos é divisível por 11. Por exemplo, 3443 é divisível por 11, já que $3 - 4 + 4 - 3 = 0$ é divisível por 11.

Os critérios de divisibilidade por 2 e 5 são fáceis demais e não precisamos de congruências para entender porque são verdade. Com isto temos critérios de divisibilidade simples de executar para todos os primos de 2 a 11, menos para 7. Vamos ver o que acontece quando testamos divisibilidade por 7.

Já sabemos que a etapa crucial das contas, e que difere em cada caso, é o cálculo do resto das potências de 10 pelo primo. Se o primo é 7, temos $10 \equiv 3 \pmod 7$; o que já não é tão simples quanto nos casos anteriores. O que acontece com outras potências maiores de 10? Vejamos alguns exemplos. Todas as congruências abaixo são módulo 7:

$$\begin{aligned}
10^2 &\equiv 3^2 \equiv 2\\
10^3 &\equiv 10 \cdot 10^2 \equiv 3 \cdot 2 \equiv 6 \equiv -1\\
10^4 &\equiv 10 \cdot 10^3 \equiv (-1) \cdot 3 \equiv 4\\
10^5 &\equiv 10 \cdot 10^4 \equiv 3 \cdot 4 \equiv 5\\
10^6 &\equiv 10 \cdot 10^5 \equiv 3 \cdot 5 \equiv 1.
\end{aligned}$$

Daí em diante não adianta continuar, porque os restos vão se repetir. Agora o problema está claro. O único resto possível de uma potência de 10 na divisão por 3 é 1; já na divisão por 11, dois restos podem ocorrer 1 e -1. Mas na divisão por 7 há 6 restos diferentes que podem ocorrer. Isto torna o critério de divisibilidade por 7 meio inútil, de tão difícil de lembrar que é. De qualquer forma vamos ver como seria o critério para números de 3 algarismos. Se a tem 3 algarismos, então $a = a_2 10^2 + a_1 10 + a_0$. Usando os restos de potências de 10 por 7 calculados acima, obtemos

$$a \equiv a_2 10^2 + a_1 10 + a_0 \equiv 2a_2 + 3a_1 + a_0 \pmod 7.$$

Portanto a é divisível por 7 quando $2a_2 + 3a_1 + a_0$ é divisível por 7. Por exemplo, 231 é divisível por 7 porque $2 \cdot 2 + 3 \cdot 3 + 1 = 14$ é divisível por 7. É claro que quanto mais algarismos tiver a, mais complicado fica o critério de divisibilidade por 7.

5. Potências

Uma outra aplicação importante das congruências é no cálculo de restos da divisão de uma potência por um número qualquer. Comecemos com um exemplo. Queremos calcular o resto da divisão de 10^{135} por 7. Vimos acima que $10^6 \equiv 1 \pmod 7$. Dividindo 135 por 6 temos $135 = 6 \cdot 22 + 3$. Temos então as seguintes congruências módulo 7

$$10^{135} \equiv (10^6)^{22} \cdot 10^3 \equiv (1)^{22} \cdot 10^3 \equiv 6.$$

Logo o resto da divisão de 10^{135} por 7 é 6.

Nem sempre é tão fácil assim. Por exemplo, qual o resto da divisão de 3^{64} por 31? Calculando os restos das potências de 3 encontramos $3^3 \equiv -4$

(mod 31). Ao invés de continuar tentando descobrir uma potência de 3 cujo resto na divisão por 31 dá 1, vamos usar o resultado que já obtivemos. Como $64 = 3 \cdot 21 + 1$, obtemos as seguintes congruências módulo 31,

$$3^{64} \equiv (3^3)^{21} \cdot 3 \equiv (-4)^{21} \cdot 3 \equiv -(2)^{42} \cdot 3.$$

Ainda não chegamos onde queríamos, mas o problema já ficou mais fácil. De fato $2^5 \equiv 1 \pmod{31}$. Voltando as congruências anteriores, e usando que $42 = 8 \cdot 5 + 2$, temos

$$3^{64} \equiv -(2)^{42} \cdot 3 \equiv -(2^5)^8 \cdot 2^2 \cdot 3 \equiv -12 \pmod{31}.$$

Como $-12 \equiv 19 \pmod{31}$, então o resto da divisão de 3^{64} por 31 é 19. Só por curiosidade, experimente determinar qual o menor expoente r tal que $3^r \equiv 1 \pmod{31}$.

Um ponto merece uma advertência especial. Considere o seguinte exemplo: qual o resto de 6^{35} por 16? Não adianta tentar encontrar qual a menor potência de 6 cujo resto por 16 é 1: esta potência não existe. De fato

$$6^4 \equiv 2^4 \cdot 3^4 \equiv 0 \cdot 3^4 \equiv 0 \pmod{16}.$$

Assim

$$6^{35} \equiv 6^4 \cdot 6^{31} \equiv 0 \pmod{16}.$$

Estes exemplos são ilustrativos de uma miscelânea de truques que usamos para calcular restos da divisão de potências por inteiros dados. Só há uma maneira para saber qual o melhor truque a ser usado: prática. É claro que um computador geralmente não usa estes truques; calcular os restos acima em um sistema de computação algébrica não leva sequer um segundo.

6. Equações diofantinas

Nossa próxima aplicação de congruências será à resolução de equações diofantinas. Uma *equação diofantina* é uma equação em várias incógnitas com coeficientes inteiros. Por exemplo, $3x - 2y = 1$, $x^3 + y^3 = z^3$ e $x^3 - 117y^3 = 5$ são equações diofantinas. Estaremos interessados em encontrar as soluções inteiras destas equações. Este tipo de problema foi abordado pelo matemático grego Diofanto em seu tratado *Aritmética*, escrito por volta de 250 d. C.. É por isso que equações deste tipo são chamadas de diofantinas.

Como as equações têm várias variáveis, pode haver uma infinidade de soluções inteiras. Por exemplo, qualquer que seja k inteiro, os números $x = 1 + 2k$ e $y = 1 + 3k$ satisfazem a equação $3x - 2y = 1$. Já a equação $x^3 + y^3 = z^3$ é um caso particular do *Último Teorema de Fermat*, que já

encontramos ao final do capítulo 2 e na introdução. Como vimos, se três inteiros x, y e z satisfazem esta equação, então $x \cdot y \cdot z = 0$. Para o caso especial da equação acima isto já havia sido provado pelo matemático suíço L. Euler em 1770. Lembre-se que foi na margem de sua cópia da *Aritmética* de Diofanto que Fermat enunciou o Último Teorema.

Já a equação $x^3 - 117y^3 = 5$ tem uma história mais recente. Em um artigo publicado em 1969, D.J. Lewis afirmava que esta equação tem, no máximo, 18 soluções inteiras. Dois anos depois R. Finkelstein e H. London provaram que a equação não tem, de fato, nenhuma solução inteira. A demonstração não é muito complicada; ocupa apenas a página 111 do volume 14 do *Canadian Mathematical Bulletin*. Ainda assim, os métodos utilizados não são normalmente estudados no curso de graduação em matemática. Entretanto, em 1973, F. Halter-Koch e V. Şt. Udresco observaram, independentemente, que existe uma maneira muito fácil de verificar que esta equação não tem solução; basta usar congruência módulo 9. Vamos fazer isto em detalhes.

Digamos que a equação $x^3 - 117y^3 = 5$ tem solução inteira. Isto significa que existem inteiros x_0 e y_0 tais que $x_0^3 - 117y_0^3 = 5$. Esta última é uma relação entre números inteiros, logo podemos reduzi-la módulo 9. Como 117 é divisível por 9, obtemos

$$x_0^3 \equiv x_0^3 - 117y_0^3 \equiv 5 \pmod 9.$$

Logo, se a equação dada tem soluções inteiras x_0 e y_0 então $x_0^3 \equiv 5 \pmod 9$. Mas será que isto é possível? As classes distintas módulo 9 são representadas pelos números de 0 a 8. Portanto basta calcular o cubo módulo 9 de cada um destes números para ver se algum deles é congruente a 5:

classes módulo 9:	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{5}$	$\overline{6}$	$\overline{7}$	$\overline{8}$
cubos módulo 9:	$\overline{0}$	$\overline{1}$	$\overline{8}$	$\overline{0}$	$\overline{1}$	$\overline{8}$	$\overline{0}$	$\overline{1}$	$\overline{8}$

Logo, o cubo de qualquer inteiro é congruente, módulo 9, a 0, 1 ou 8. Em particular $x_0^3 \equiv 5 \pmod 9$ não tem solução. Portanto a equação $x^3 - 117y^3 = 5$ não tem soluções inteiras.

A história deste exemplo pede que dela se tire uma moral: *freqüentemente a primeira demonstração de um teorema não é a mais simples, nem a mais elegante*. A primeira demonstração é geralmente obtida pelo desbravador, aquele que vai abrindo caminho. Com o passar do tempo vai ficando clara a verdadeira posição 'geográfica' da descoberta: onde se situa e como está relacionada a outros fatos conhecidos. Com isto se torna possível obter uma demonstração mais simples e mais direta. Naturalmente o exemplo acima é um caso extremamente ingênuo; nem por isso o fato geral é menos verdadeiro.

O matemático russo A. S. Besicovitch dizia, a este respeito: *'A reputação de um matemático é proporcional ao número de demonstrações ruims que ele deu'*.

7. Divisão modular

É hora de voltarmos à questão de como calcular divisões em $\mathbb{Z}_n$. Digamos que a e b são números reais. A frase 'dividir a por b' é equivalente a 'multiplicar a por $1/b$'. O número real $1/b$ é conhecido como o *inverso* de b. O que caracteriza $1/b$ é a equação $b \cdot 1/b = 1$. Isto pode parecer meio circular. Entretanto, trabalhar com inversos ao invés de falar diretamente em divisão, tem uma vantagem conceitual que ficará clara mais adiante. Se você faz o estilo atento, deve ter notado que precisamos assumir que $b \neq 0$ para as afirmações acima fazerem sentido. Isto é, um número real só tem inverso se é diferente de zero. Vamos transpor tudo isto para $\mathbb{Z}_n$.

Digamos que $\overline{a} \in \mathbb{Z}_n$. Diremos que a classe $\overline{\alpha} \in \mathbb{Z}_n$ é o *inverso* de $\overline{a}$ se a equação $\overline{a} \cdot \overline{\alpha} = \overline{1}$ é verificada em $\mathbb{Z}_n$. É claro que se $\overline{a} = \overline{0}$ então $\overline{a}$ não tem inverso. Infelizmente em $\mathbb{Z}_n$ pode haver outros elementos sem inverso além da classe $\overline{0}$.

Suponhamos que $\overline{a} \in \mathbb{Z}_n$ tem inverso $\overline{\alpha}$ e vejamos aonde chegamos com isto. A equação

$$\overline{a} \cdot \overline{\alpha} = \overline{1}$$

corresponde a dizer que $a\alpha - 1$ é divisível por n. Isto é

$$a\alpha + kn = 1 \tag{7.1}$$

para algum inteiro k. Observe que esta última equação implica que $\text{mdc}(a, n) = 1$. Concluímos portanto que se $\overline{a}$ tem inverso em $\mathbb{Z}_n$ então $\text{mdc}(a, n) = 1$. Será que a recíproca é verdadeira?

Suponhamos então que a é um inteiro e que $\text{mdc}(a, n) = 1$. A equação (7.1) sugere que apliquemos o algoritmo euclidiano estendido aos números a e n para obter inteiros α e β tais que

$$a\alpha + n\beta = 1.$$

Esta equação é equivalente a

$$\overline{a} \cdot \overline{\alpha} = \overline{1}$$

em $\mathbb{Z}_n$. Logo a classe $\overline{\alpha}$ calculada pelo algoritmo euclidiano estendido é o inverso de a em $\mathbb{Z}_n$. Concluímos assim que se $\text{mdc}(a, n) = 1$ então $\overline{a}$ tem inverso em $\mathbb{Z}_n$. Vamos resumir tudo isto em um teorema.

Teorema de inversão. *A classe $\overline{a}$ tem inverso em $\mathbb{Z}_n$ se, e somente se, a e n são primos entre si.*

Nossa demonstração faz mais, porque nos diz como calcular o inverso quando ele existe. Basta usar o algoritmo euclidiano estendido. Por exemplo: qual o inverso de $\overline{3}$ em $\mathbb{Z}_{32}$? A primeira pergunta é se o inverso existe mesmo. Como vamos aplicar o algoritmo estendido para calculá-lo descobrimos *se* o inverso existe e *qual é* ao mesmo tempo. No exemplo, é claro que $\text{mdc}(3, 32) = 1$; portanto o inverso existe. Aplicando o algoritmo estendido obtemos

$$3 \cdot 11 - 32 = 1,$$

que é equivalente a $\overline{3} \cdot \overline{11} = \overline{1}$ em $\mathbb{Z}_{32}$. Logo $\overline{11}$ é o inverso de $\overline{3}$ em $\mathbb{Z}_{32}$. É claro que qual é o inverso e até mesmo sua existência depende de modo crucial do valor do módulo. Por exemplo, a classe $\overline{3}$ em $\mathbb{Z}_{81}$ *não* tem inverso.

O conjunto dos elementos de $\mathbb{Z}_n$ que têm inverso é muito importante. Vamos denotá-lo por $U(n)$. Em símbolos

$$U(n) = \{\overline{a} \in \mathbb{Z}_n : \text{mdc}(a, n) = 1\}.$$

É fácil calcular $U(p)$ quando p é um número primo. Neste caso $\text{mdc}(a, p) = 1$ significa que p não divide a. Mas se p divide a então $\overline{a} = \overline{0}$. Portanto, quando p é primo, todas as classes diferentes de $\overline{0}$ têm inverso. Assim

$$U(p) = \mathbb{Z}_p \setminus \{\overline{0}\}.$$

Mas isto vale *apenas quando p é primo.* O que acontece quando n é um número composto é ilustrado nos seguintes exemplos:

$$U(4) = \{\overline{1}, \overline{3}\} \quad \text{e} \quad U(8) = \{\overline{1}, \overline{3}, \overline{5}, \overline{7}\}.$$

Uma propriedade importante do conjunto $U(n)$ é que o produto de dois elementos de $U(n)$ é um elemento de $U(n)$. Em outras palavras, se $\overline{a}$ e $\overline{b}$ em $\mathbb{Z}_n$ têm inverso, então $\overline{a} \cdot \overline{b}$ também tem inverso em $\mathbb{Z}_n$. Isto será tão importante no futuro que vamos verificar com detalhes. Digamos que $\overline{a}$ tem inverso $\overline{\alpha}$ e $\overline{b}$ tem inverso $\overline{\beta}$ em $\mathbb{Z}_n$. O inverso de $\overline{a} \cdot \overline{b}$ será–surpresa! surpresa!–$\overline{\alpha} \cdot \overline{\beta}$. A verificação é imediata

$$(\overline{a} \cdot \overline{b})(\overline{\alpha} \cdot \overline{\beta}) = (\overline{a} \cdot \overline{\alpha})(\overline{b} \cdot \overline{\beta}) = \overline{1} \cdot \overline{1} = \overline{1}.$$

Teremos muito mais a dizer sobre $U(n)$ em capítulos futuros. Em particular vamos estar interessados em determinar o número de elementos de $U(n)$ em função de n, e em descobrir quando $U(n)$ tem uma *raiz primitiva*. Uma classe $\overline{a}$ de $U(n)$ é uma *raiz primitiva* se todo elemento de $U(n)$ é igual

a uma potência de $\overline{a}$. Por exemplo $\overline{3}$ é uma raiz primitiva de $U(7)$. De fato, em $\mathbb{Z}_7$ temos que

$$\begin{aligned}
\overline{3}^2 &= \overline{2} \\
\overline{3}^3 &= \overline{3} \cdot \overline{2} = \overline{6} \\
\overline{3}^4 &= \overline{3} \cdot \overline{6} = \overline{4} \\
\overline{3}^5 &= \overline{3} \cdot \overline{4} = \overline{5} \\
\overline{3}^6 &= \overline{3} \cdot \overline{5} = \overline{1}.
\end{aligned}$$

Contudo $\overline{2}$ não é uma raiz primitiva de $U(7)$; a verificação fica a seu cargo.

Finalmente voltemos à questão de executar divisões em $\mathbb{Z}_n$. Se queremos dividir $\overline{a}$ por $\overline{b}$ precisamos saber se $\overline{b}$ está ou não em $U(n)$. Se não estiver, a divisão não é possível. Se estiver, calculamos o inverso de $\overline{b}$ em $\mathbb{Z}_n$ (digamos que é $\overline{\beta}$) e dividimos $\overline{a}$ por $\overline{b}$ calculando o produto $\overline{a} \cdot \overline{\beta}$. Um exemplo: calcular a divisão de $\overline{2}$ por $\overline{3}$ em $\mathbb{Z}_8$. Como $\text{mdc}(3, 8) = 1$ então $\overline{3}$ tem inverso em $\mathbb{Z}_8$. Neste caso os números são tão pequenos que fica fácil achar o inverso por tentativa. O inverso de $\overline{3}$ em $\mathbb{Z}_8$ é o próprio $\overline{3}$. Assim o resultado da divisão de $\overline{2}$ por $\overline{3}$ em $\mathbb{Z}_8$ é $\overline{6}$. Observe que o fato de $\overline{2}$ não ter inverso em $\mathbb{Z}_8$ não tem nenhuma importância neste caso, já que $\overline{2}$ é o dividendo.

Podemos usar o que aprendemos nesta seção para resolver congruências lineares em $\mathbb{Z}_n$. Uma *congruência linear* é uma equação do tipo

$$ax \equiv b \pmod{n}, \tag{7.2}$$

onde $a, b \in \mathbb{Z}$. Para resolvê-la, precisamos 'dividir' por a para deixar o x livre do lado esquerdo da equação. É claro que isto só é possível se $\text{mdc}(n, a) = 1$. Digamos que esta última hipótese é satisfeita. Então existe $\alpha \in \mathbb{Z}$ tal que $\alpha a \equiv 1 \pmod{n}$. Multiplicando a equação (7.2) por α, obtemos $\alpha a x \equiv \alpha b \pmod{n}$. Como $\overline{\alpha}$ é o inverso de $\overline{a}$ em $\mathbb{Z}_n$ esta equação se reduz a

$$x \equiv \alpha b \pmod{n}.$$

Por exemplo, para resolver a congruência $7x \equiv 3 \pmod{15}$, precisamos multiplicá-la pelo inverso de $\overline{7}$ em $\mathbb{Z}_{15}$. Como $15 - 2 \cdot 7 = 1$, o inverso de $\overline{7}$ é $\overline{-2} = \overline{13}$. Multiplicando a congruência por 13, temos

$$x \equiv 13 \cdot 3 \equiv 39 \equiv 9 \pmod{15}.$$

Achamos assim a solução da equação.

Uma conclusão importante a que chegamos, contemplando este método de resolução de congruências, é a seguinte. Se $\text{mdc}(a, n) = 1$, então a congruência linear $ax \equiv b \pmod{n}$ tem uma, e uma só, solução em $\mathbb{Z}_n$. Isto pode não ser verdadeiro se eliminamos a condição $\text{mdc}(a, n) = 1$. Por exemplo, a equação $2x \equiv 1 \pmod{8}$ *não* tem solução. Voltaremos a esta questão no capítulo 6.

8. Exercícios

1. Considere as relações abaixo, definidas no conjunto $\mathbb{Z}$ dos números inteiros. Determine se são reflexivas, simétricas ou transitivas. Alguma das duas relações é de equivalência?
(1) $a \sim b$ quando $\text{mdc}(a, b) = 1$.
(2) Fixe $n > 0$ inteiro. Então $a \sim b$ quando $\text{mdc}(a, n) = \text{mdc}(b, n)$.

2. Para cada par de inteiros a, n abaixo ache um número inteiro b tal que $b \equiv a \pmod{n}$ e $0 \leqslant b < n$.
(1) $a = 2351$ e $n = 2$
(2) $a = 50121$ e $n = 13$
(3) $a = 321671$ e $n = 14$

3. Calcule:
(1) $5^{20} \pmod{7}$
(2) $7^{1001} \pmod{11}$
(3) $2^{130} \pmod{263}$
(4) $13^{221} \pmod{19}$

4. Calcule o resto da divisão de $1000!$ por 3^{300}.

5. Quais os elementos de $\mathbb{Z}_4$ que têm inversos? E de $\mathbb{Z}_{11}$? E de $\mathbb{Z}_{15}$? Calcule os inversos desses elementos em cada caso.

6. Resolva as seguintes equações:
(1) $4x \equiv 3 \pmod{4}$.
(2) $3x + 2 \equiv 0 \pmod{4}$.
(3) $2x - 1 \equiv 7 \pmod{15}$

7. Ache um elemento $\overline{a}$ de $\mathbb{Z}_{34}$ de modo que todo elemento invertível de $\mathbb{Z}_{34}$ é uma potência de $\overline{a}$.

8. Mostre que a equação diofantina $x^2 - 7y^2 = 3$ não tem soluções inteiras.

9. Mostre que $p = 274177 = 1071 \cdot 2^8 + 1$ é fator primo do número de Fermat $F(6)$.

Sugestão: Comece calculando 1071^8 módulo p. Para fazer isto é melhor usar que $1071 = 7 \cdot 9 \cdot 17$ e calcular a oitava potência de cada fator módulo p separadamente. Como $p = 1071 \cdot 2^8 + 1$ temos que $(1071 \cdot 2^8)^8 \equiv 1 \pmod{p}$. Por outro lado $(1071 \cdot 2^8)^8 \equiv 1071^8 \cdot 2^{64} \pmod{p}$. Substitua o valor de 1071^8 módulo p nesta última fórmula e compare com a congruência anterior. O fato de p dividir $F(6)$ sai disso como por mágica!

10. O objetivo desta questão é mostrar que nenhum número da forma $4n+3$ pode ser escrito como a soma dos quadrados de dois inteiros.

(1) Mostre que o quadrado de qualquer inteiro só pode ser congruente a 0 ou 1 módulo 4.

(2) Use (1) para mostrar que se x e y são inteiros então $x^2 + y^2$ só pode ser congruente a 0, 1 ou 2 módulo 4.

(3) Use (2) para mostrar que um inteiro da forma $4n+3$ não pode ser escrito como soma de dois quadrados de inteiros.

Este resultado é um caso particular de um teorema comunicado por Fermat em uma carta a Roberval datada de 1640. Fermat também sabia que qualquer primo da forma $4n + 1$ pode ser escrito como soma de dois quadrados de inteiros. Para mais detalhes consulte [**Weil 1987, Cap. II,** §**VIII**]. Veja também o exercício 12 capítulo 5.

11. Escreva um programa que implemente o algoritmo para calcular potências módulo n que é descrito na seção 2 do apêndice. O algoritmo deverá ter como entrada a, k e n, onde a é um inteiro qualquer e k e n são inteiros positivos. A saída deverá ser a forma reduzida de a^k módulo n. Este algoritmo é uma parte fundamental de quase todos os algoritmos que estudaremos a partir do capítulo 6.

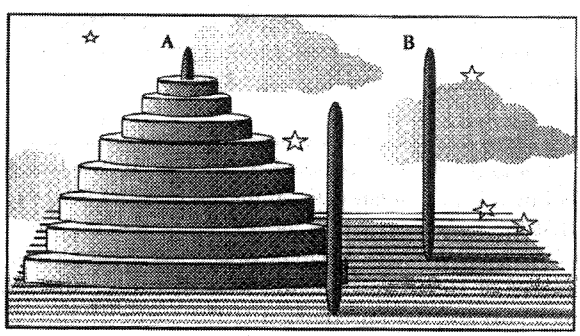
5

Indução e Fermat

Estamos prontos para retomar o estudo dos números primos, iniciado anteriormente. Quase tudo o que faremos daqui por diante dependerá do *Pequeno Teorema de Fermat*, que é o principal resultado deste capítulo. Há muitas maneiras de provar este resultado, a mais simples faz uso do método de *indução*. No capítulo que trata de grupos, obteremos o mesmo teorema como conseqüência de um resultado muito mais abrangente. O teorema de Fermat vai ser o pivô da maioria de nossas aplicações daqui por diante.

1. Hanói! Hanói!

Será que, quando criança, você brincava com um jogo chamado *Torres de Hanói*? O jogo consiste em uma base de madeira onde estão firmadas três hastes verticais, e em um certo número de discos de madeira (seis no meu jogo), de diâmetros diferentes, furados no centro. Vamos chamar de **A**, **B** e **C**, as três hastes, conforme mostra a figura.



No começo do jogo os discos estão todos enfiados na haste **A**, em ordem decrescente de tamanho, com o menor disco acima de todos. O objetivo é mover todos os discos, de **A** para **C**, obedecendo às seguintes regras:
(1) Somente um disco pode ser movido de cada vez.

(2) Um disco maior nunca pode ser posto sobre um disco menor.

É claro que isto só é possível usando a haste **B** como intermediário. O problema que proporemos vai um pouco mais além: *qual o número mínimo de movimentos que precisaremos fazer para alcançar o objetivo?*

Para os que gostam de encontrar uma aplicação prática para tudo, aqui vai uma história. Em Benares, no norte da Índia, sob a grande cúpula do templo de Brama, haveria um enorme jogo de Torre de Hanói, com hastes de diamante e 64 discos de ouro. O jogo teria sido posto lá pelo deus Brama em pessoa, no momento da criação do mundo. Naquele momento os discos estariam todos empilhados em uma das hastes. Caberia aos sacerdotes mover os discos, segundo as regras acima. No momento em que todos os discos tiverem sido empilhados na outra haste, terá chegado a hora do deus voltar e pôr fim ao mundo com um relâmpago. *Portanto*, para saber a data do fim do mundo precisamos resolver o problema posto acima. Quer aplicação mais prática?

Voltando a questões menos escatológicas, consideremos o jogo de madeira, com o qual iniciamos a seção. Se o jogo só tivesse um disco, seria fácil movê-lo (segundo as regras!) de **A** para **C**. Para isso precisamos de apenas um movimento. Vamos considerar o caso de dois discos. Movemos o disco menor para **B**; o segundo disco para **C** e depois o menor de **B** para **C**: acabou. Fizemos 3 movimentos. Seria bom que você brincasse um pouco com o jogo de seis discos antes de continuar a ler.

Pausa para brincar com o jogo.

Vamos tratar agora do caso geral de um jogo com n discos. Considere o seguinte diálogo:

Professor: Vamos imaginar que os discos tenham sido numerados de cima para baixo: $1, 2, \ldots, n$. O menor disco é o 1, e o maior (e mais de baixo) é o n. O que é preciso fazer para remover o disco n?

Aluno: Uh?

Professor: Para chegar a mover o disco n, o que é que eu tenho que fazer?

Aluno: Tirar todos os de cima?

Professor: Pois é, tenho que tirar todos os $n-1$ discos que estão acima dele. Lembre-se que quero mover os discos todos para a haste **C**, e o disco n é o que deve ficar mais embaixo nesta haste. Onde seria preferível colocar os outros discos?

Aluno: Na haste **B**?

Professor: Na haste **B**. Só há um problema...

Aluno: Sempre há.

Professor: As regras. Pela regra 1 só posso mover um disco de cada vez; pela regra 2 os $n-1$ discos têm que estar empilhados em ordem decrescente de tamanho na haste **B**. O que tenho que fazer para mover os $n-1$ discos menores, de **A** para **B**?
Aluno: Mover um de cada vez, respeitando as regras?

Professor: Mais precisamente?
Aluno: Deve ser como jogar com $n-1$ peças, só que empilhando na haste **B**, em vez de **C**.

Professor: É isso. Resumindo: para remover o disco n para **C**, temos que remover os $n-1$ acima dele para **B**. Feito isto removemos o disco n para **C**.
Aluno: Como é que os $n-1$ discos vão para **C**? Quer dizer, só se for repetindo todo o jogo, de modo a passar todo mundo (um a um?) de **B** para **C**.

Professor: É demorado, mas é isso. Observe que temos que fazer o jogo com $n-1$ discos *duas vezes*: primeiro movemos os $n-1$ discos de **A** para **B** (usando **C** como intermediário, claro). Isto descobre o disco n. Movemos então n para **C**. Agora jogamos com os $n-1$ discos mais uma vez: de **B** para **C**, usando **A** como intermediário. Com isto empilhamos todos em **C** sem violar as regras.
Aluno: Uff!

Professor: Ainda não acabou.
Aluno: Não?!

Professor: Queremos saber qual o número mínimo de movimentos, não queremos?
Aluno: Acho que ainda queremos...

Professor: Para facilitar, vamos dizer que o número *mínimo* de movimentos necessários para completar o jogo de n discos é $T(n)$. Agora, não há como chegar ao disco n sem mover os $n-1$ de cima: quantos movimentos fizemos para isso?
Aluno: $T(n-1)$?

Professor: Como, espertamente, movemos os $n-1$ para a haste **B**, a haste **C** está livre. Logo podemos mover o disco n para **C**. Quantos movimentos até agora?
Aluno: Um?

Professor: Quero dizer, quantos movimentos desde que começamos o jogo?
Aluno: $T(n-1)+1$?

Professor: O que falta?
Aluno: Mover os $n-1$ discos de **B** para **C**, para ficarem em cima do disco n.

Professor: Qual o número mínimo de movimentos para fazer isto?
Aluno: $T(n-1)$?
Professor: Logo, desde o começo do jogo fizemos $T(n-1)+1+T(n-1) = 2T(n-1)+1$ movimentos. Além disso, a análise que fizemos do jogo mostra que não é possível fazer um número menor de movimentos. Então quem é $T(n)$?
Aluno: É o menor número de movimentos para completar o jogo de n discos.
Professor: É, mas como calculamos $T(n)$ se sabemos $T(n-1)$?
Aluno: $T(n) = 2T(n-1)+1$?
Professor: Pois é. Com isso fica fácil calcular o número mínimo de movimentos para o jogo de 6 discos.

O resultado final do diálogo acima é a fórmula $T(n) = 2T(n-1)+1$. Observe que esta fórmula não nos dá o valor de $T(n)$ diretamente, mas apenas em função do valor anterior $T(n-1)$. Para achar $T(n)$ precisamos aplicar a fórmula repetidas vezes. Por exemplo, para calcular $T(6)$ precisamos antes achar $T(1), T(2), \ldots, T(6)$. Já vimos que $T(1) = 1$. Logo

$$T(2) = 2T(1)+1 = 3.$$

Continuando assim, temos que

$$T(3) = 7,\ T(4) = 15,\ T(5) = 31 \text{ e } T(6) = 63.$$

Portanto, temos que fazer pelo menos 63 movimentos para mover todos os discos de **A** para **C** na minha Torre de Hanói. E no problema posto por Brama em Benares? Para resolvê-lo é necessário calcular $T(64)$, uma tarefa meio ingrata.

Uma fórmula como $T(n) = 2T(n-1)+1$ é chamada de *fórmula de recorrência*; afinal para calcular $T(n)$ precisamos aplicar a fórmula repetidas vezes. Talvez você esteja se perguntando: *como se prova esta fórmula de recorrência?* Na verdade o diálogo acima é uma demonstração da fórmula, embora expressa de uma maneira pouco convencional. Experimente, como exercício, extrair as etapas essenciais do diálogo e reescrever a demonstração da maneira usual.

O fato da fórmula obtida ser de recorrência não nos impede de sonhar com a existência de uma fórmula que nos dê $T(n)$ diretamente a partir de n. Volte a olhar os números acima. Não é difícil adivinhar uma fórmula para $T(n)$ a partir deles. A fórmula é fácil e simples, vale a pena tentar encontrá-la. Uma vez advinhada a fórmula, precisamos mostrar que é verdadeira sempre,

e não apenas para os números tabelados acima. Para fazer isto precisaremos de um novo método de demonstração, a *indução finita*.

Talvez você esteja pensando: *mas para quê outra fórmula? qual o problema com a recorrência?* Afinal para aplicá-la só precisamos multiplicar e somar muitos números, isto é uma coisa que um computador faz muito rápido. Este é um ponto em que a computação e a matemática atuam como forças opostas. Caricaturizando um pouco, podemos dizer que a computação procura processar os métodos de *força bruta* de forma mais rápida. Já a matemática almeja substituí-los por maneiras indiretas de proceder, que produzam o resultado com um mínimo de cálculos. É claro que as duas são, na verdade, faces de uma mesma moeda.

2. Indução finita

A palavra *indução* é usada em matemática em um sentido técnico, às vezes qualificado pelo adjetivo *finita*, como no título da seção. Mas a palavra também tem um uso vulgar. Segundo o Aurélio, *indução* é

a operação mental que consiste em se estabelecer uma verdade universal ou uma proposição geral com base no conhecimento de certo número de dados singulares ou de proposições de menor generalidade.

Quando você advinhou a fórmula de $T(n)$ na seção acima, a partir dos valores de $T(1)$ a $T(6)$, você estava praticando uma indução. É claro que fazemos isto todo o tempo no dia a dia. Mas em matemática isto pode ser perigoso (no dia a dia também!).

Um exemplo notório de indução descarrilhada deve-se a Fermat. Já vimos que em uma carta de 1642, Fermat havia afirmado que os números da forma

$$2^{2^n} + 1$$

são sempre primos. Acredita-se que Fermat teria chegado a esta conclusão calculando estes números para $n = 0, \ldots, 4$. De fato, neste caso temos os números:

$$3,\ 5,\ 17,\ 257 \text{ e } 65537;$$

que são todos primos. O número seguinte já é bem maior:

$$2^{2^5} + 1 = 4294967297.$$

Não se sabe porque Fermat não tentou fatorar este número; por causa do tamanho, talvez? A verdade é que este número é composto, como mostrou Euler em 1738. Voltaremos a esta questão no capítulo 9. Por enquanto a história serve apenas como ilustração dos perigos da indução em matemática.

Já na época de Fermat os matemáticos preocupavam-se com estas *induções* ou *generalizações mal feitas*. Tanto Fermat, quanto seu contemporâneo B. Pascal, conheciam o *método de indução finita*, também conhecido por *indução matemática* ou *demonstração por recorrência*. Este é o mesmo Pascal que fez experiências importantes com o barômetro e projetou e construiu uma das primeiras máquinas mecânicas de calcular. Mas ele é ainda mais famoso nos meios literários, como autor dos *Pensamentos*.

Voltando à indução finita. Digamos que temos uma *proposição* que queremos provar, e que é do seguinte tipo *'tal coisa vale para todos os números naturais'*. Por exemplo: no jogo das Torres da Hanói com n discos o número mínimo de movimentos é $2^n - 1$. Ou ainda, se p é um número primo então $n^p - n$ é divisível por p, qualquer que seja o número natural n. A matemática está cheia de proposições deste tipo. Em geral, podemos imaginar que temos uma proposição $P(n)$ que afirma que uma determinada propriedade vale para cada número natural n.

Princípio de indução finita. *Seja* $P(n)$ *a proposição que queremos provar. Para que* $P(n)$ *seja verdadeira para todo* n *natural, basta que:*
(1) $P(1)$ *seja verdadeira;*
(2) Se $P(k)$ *for verdadeira para algum número natural* k, *então* $P(k+1)$ *também seja verdadeira.*

Não vamos demonstrar que o método funciona. Para apreciar a demonstração é necessário ter conhecimento das propriedades básicas (axiomas) dos números naturais. Tirar um dos axiomas de contexto só para provar o método de indução levaria a uma sensação indesejável de desconforto. Ao invés disto, vamos nos convencer de que o método é razoável. Começamos sabendo que $P(1)$ é verdade. Mas, de acordo com (2), se $P(1)$ é verdade, então $P(2)$ é verdade. Por (2) de novo, se $P(2)$ é verdade, então $P(3)$ é verdade; e assim por diante.

É como a queda de um dominó. Conhece a brincadeira? Ponha cada peça do dominó, de pé, do lado de outra, a uma pequena distância. Neste caso, a afirmação $P(n)$ é: *a* n*-ésima peça do dominó cai*. De fato, se derrubamos a primeira peça (isto é, se $P(1)$ é verdadeira) e se a queda da k-ésima peça derruba a $k+1$-ésima (isto é, se $P(k)$ verdadeira implica que $P(k+1)$ é verdadeira) então caem todas as peças do dominó.

Vejamos um outro exemplo, o jogo das Torres de Hanói. Você deve ter adivinhado $T(n) = 2^n - 1$. Como obtivemos a fórmula a partir de alguns dados numéricos (isto é, por *indução vulgar*) queremos saber se é mesmo verdadeira. Para isso vamos usar *indução finita*. Já vimos que $T(1) = 1$.

Como $2^1 - 1 = 1$, a fórmula vale neste caso. Suponhamos que $T(n) = 2^n - 1$, queremos mostrar que $T(n+1) = 2^{n+1} - 1$. Para realizar esta etapa da demonstração precisamos de dois ingredientes:

(1) A *hipótese de indução*: isto é, a suposição de que a proposição vale para n. Neste caso a hipótese de indução é $T(n) = 2^n - 1$.

(2) Uma maneira de relacionar $T(n)$ a $T(n+1)$. Neste caso, temos a lei de recorrência $T(n+1) = 2T(n) + 1$, obtida na seção 1.

Começamos com $T(n+1) = 2T(n) + 1$. Como, pela hipótese de indução, $T(n) = 2^n - 1$, podemos substituir isto na primeira fórmula para obter

$$T(n+1) = 2T(n) + 1 = 2(2^n - 1) + 1 = 2^{n+1} - 1,$$

que era o resultado esperado. Logo a fórmula $T(n) = 2^n - 1$ vale para qualquer n natural. Pedante? Neste exemplo talvez, mas à medida que as proposições vão ficando mais complicadas, fica difícil viver sem indução. Um bom exemplo está na próxima seção.

Tendo achado a fórmula para $T(n)$ podemos calcular a data do fim do mundo – ao menos segundo a história de Brama. O número de movimentos que os sacerdotes terão que fazer com os discos de ouro é $2^{64} - 1$, que é da ordem de 10^{19}. Para poder transformar isto em unidades de tempo, digamos que o movimento de um dos discos de ouro toma meia-hora, em média. Fazendo as contas vemos que isto dá cerca de 10^{14} anos do início ao fim do mundo. Como, acredita-se, 10^{11} anos passaram-se desde o Big Bang, ainda temos muito tempo até que chegue a data do fim do mundo. Pensando bem, Brama não poderia ter posto o jogo na Índia antes da criação da Terra, mas não vamos nos preocupar com detalhes.

3. Pequeno Teorema de Fermat

O *Pequeno Teorema de Fermat* afirma que se p é um número primo e a um inteiro qualquer então p divide $a^p - a$. Alguns casos deste teorema já eram conhecidos desde a antigüidade. Por exemplo, os chineses sabiam que se p é primo então p divide $2^p - 2$. Mas foi Fermat quem obteve o resultado geral e o introduziu na matemática européia do século XVII. A maneira moderna de enunciar o teorema faz uso da linguagem de congruências.

Teorema de Fermat. *Seja p um número primo e a um número inteiro, então*

$$a^p \equiv a \pmod{p}.$$

Vamos provar este teorema por indução finita, que foi, de fato, uma das primeiras maneiras como o teorema foi demonstrado. Para isto precisamos

achar uma proposição $P(n)$ à qual aplicar a indução. A proposição $P(n)$ será a seguinte: $n^p \equiv n \pmod{p}$. Esta afirmação é mais restritiva do que o teorema enunciado acima: a proposição só se refere a números naturais, o teorema a qualquer inteiro. Voltaremos a esta questão mais adiante.

É evidente que $P(1)$ vale, já que $1^p = 1$. Suponhamos então que $n^p \equiv n \pmod{p}$. O elo de ligação entre $P(n)$ e $P(n+1)$ é o *binômio de Newton*. É mais confortável expressar a demonstração em termos de uma versão do binômio de Newton para congruências, que vamos formular como um resultado auxiliar.

Lema. *Seja* p *um número primo e* a *e* b *inteiros. Então,*

$$(a+b)^p \equiv a^p + b^p \pmod{p}.$$

Demonstração do lema: Usando a expressão usual do binômio de Newton, temos que

$$(a+b)^p = a^p + b^p + \sum_{i=1}^{p-1} \binom{p}{i} a^{p-i} b^i.$$

Para obter o lema é suficiente mostrar que o termo

$$\sum_{i=1}^{p-1} \binom{p}{i} a^{p-i} b^i$$

é congruente a zero módulo p. Uma coisa ainda mais forte é verdade. Considere o número binomial

$$\binom{p}{i} = \frac{p(p-1)\dots(p-i+1)}{i!}.$$

Sabemos que este é um número inteiro. Para que a fração acima dê lugar a um número inteiro é preciso que o denominador seja completamente cancelado por termos no numerador. Agora suponha que $1 \leqslant i \leqslant p-1$. Então o denominador $i!$ *não* tem p como um de seus fatores primos. Assim o fator p que aparece no numerador *não* é cancelado por nenhum fator do denominador – observe que isto só é verdade porque p é primo! Portanto o número inteiro $\binom{p}{i}$ é múltiplo de p quando $1 \leqslant i \leqslant p-1$. Conseqüentemente,

$$\sum_{i=1}^{p-1} \binom{p}{i} a^{p-i} b^i \equiv 0 \pmod{p},$$

e o lema está provado.

Vamos voltar à demonstração do Teorema de Fermat. Estamos usando indução, e supondo que $n^p \equiv n \pmod{p}$. Queremos mostrar que $(n+1)^p \equiv n+1 \pmod{p}$. Usando o Lema, temos que

$$(n+1)^p \equiv n^p + 1^p \equiv n^p + 1 \pmod{p}.$$

Como, pela hipótese de indução, $n^p \equiv n \pmod{p}$, então

$$(n+1)^p \equiv n^p + 1 \equiv n + 1 \pmod{p},$$

que é o que queríamos mostrar.

Concluindo, provamos que $n^p \equiv n \pmod{p}$, para qualquer n *natural*. Mas o teorema foi enunciado para qualquer a inteiro, logo faltam os inteiros negativos. Digamos que a é um inteiro negativo. Então $-a$ é positivo: logo podemos aplicar o que já provamos a $-a$. Temos:

$$(-a)^p \equiv -a \pmod{p}.$$

Supondo que p é *ímpar*, $(-a)^p = -a^p$. Substituindo na equação acima: $-a^p \equiv -a \pmod{p}$. Multiplicando ambos os membros da congruência por -1, concluímos que $a^p \equiv a \pmod{p}$, que é o resultado do teorema. Resta-nos o caso em que $p = 2$. Este é ainda mais fácil e fica a seu cargo verificá-lo.

Provamos assim o Pequeno Teorema de Fermat. No capítulo seguinte vamos considerar algumas aplicações substanciais do teorema. Por ora nos contentaremos com aplicações mais ingênuas. Começaremos reformulando um pouco o enunciado.

Segundo o teorema, se p é primo e a é um inteiro qualquer, então $a^p \equiv a \pmod{p}$. Suponhamos que sabemos que p *não divide* a. Neste caso a é inversível módulo p pelo teorema de inversão. Seja a' um inteiro positivo tal que $aa' \equiv 1 \pmod{p}$. Multiplicando ambos os membros de $a^p \equiv a \pmod{p}$ por a', obtemos

$$a'a \cdot a^{p-1} \equiv a'a \pmod{p}.$$

Substituindo $a'a \equiv 1 \pmod{p}$ nesta equação, ficamos com

$$a^{p-1} \equiv 1 \pmod{p};$$

que é a nova versão do pequeno teorema de Fermat. Vale a pena enunciar isto claramente para uso futuro.

Teorema de Fermat II. *Seja* p *um número primo e* a *um inteiro que não é divisível por* p. *Então* $a^{p-1} \equiv 1 \pmod{p}$.

Aplicaremos esta versão do teorema ao cálculo de potências módulo p, onde p é primo. De uma maneira geral o problema é o seguinte. São dados a, k e p três inteiros positivos, dos quais sabemos que p é primo e não divide a. Digamos que k é muito grande (o caso difícil) e queremos achar a forma reduzida de a^k módulo p.

Podemos simplificar as contas usando o *teorema de Fermat*. Estamos supondo que k é grande, na prática precisamos saber apenas que $k \geqslant p-1$. Dividindo k por $p-1$ obtemos $k = (p-1)q + r$ onde o resto r satisfaz $0 \leqslant r \leqslant p-2$. Temos então que

$$a^k \equiv a^{(p-1)q+r} \equiv (a^{p-1})^q a^r \pmod{p}.$$

Mas, pelo teorema de Fermat $a^{p-1} \equiv 1 \pmod{p}$. Obtemos, portanto, da equação acima que $a^k \equiv a^r \pmod{p}$.

Um exemplo numérico para convencê-lo das vantagens deste resultado tão simples. Queremos calcular $2^{5432675}$módulo 13. Da maneira como vínhamos procedendo teríamos que efetuar uma quantidade enorme de potenciações módulo 13. Usando a idéia acima, basta calcular o resto da divisão de 5432675 por 12, que é 11. Assim

$$2^{5432675} \equiv 2^{11} \equiv 7 \pmod{13}.$$

Podia ser mais fácil?

4. Contando raízes

Há ainda uma pergunta que precisamos considerar. Será que existe algum inteiro positivo k, *menor* que $p-1$, tal que $a^k \equiv 1 \pmod{p}$ para todo inteiro a que não é divisível por p?

Uma resposta a esta questão poderia ser a seguinte. Existe um teorema bem conhecido que diz que uma equação polinomial não pode ter um número de raízes maior do que o seu grau. Acontece que se $a^k \equiv 1 \pmod{p}$, para todo inteiro a entre 1 e $p-1$, então o polinômio $x^k - 1$ tem $p-1$ raízes distintas em $\mathbb{Z}_p$. Pelo teorema mencionado acima isto só pode acontecer se $k \geqslant p-1$. Logo a resposta à questão é *não*.

Embora a resposta dada no parágrafo anterior esteja totalmente correta, ela esconde nos bastidores a sua verdadeira natureza, e as dificuldades que lhe são concomitantes. O problema é com o teorema. Usualmente, quando tratamos de equações polinomiais, procuramos raízes entre os números inteiros, racionais, reais ou complexos. Mas aqui estamos interessados em raízes que são elementos de $\mathbb{Z}_p$. É aí que reside a dificuldade. Precisamos verificar

que o teorema continua valendo neste caso. Isto não é uma questão meramente pedante: veremos que o teorema continua valendo quando o módulo é primo, mas que é *falso* quando o módulo é composto.

Teorema. *Seja* $f(x)$ *um polinômio de grau* k *com coeficientes inteiros e coeficiente líder* 1. *Se* p *é um número primo, então* $f(x)$ *não pode ter mais de* k *raízes distintas em* $\mathbb{Z}_p$.

Antes de começar a demonstração, vale a pena observar que precisamos que p seja primo porque vamos usar a propriedade: se $ab \equiv 0 \pmod{p}$ então $a \equiv 0 \pmod{p}$ ou $b \equiv 0 \pmod{p}$. É conveniente isolar o principal ingrediente da demonstração em um lema, que será provado ao final da seção.

Lema. *Seja* $h(x)$ *um polinômio com coeficientes inteiros de grau* m. *Dado um inteiro* α, *existe um polinômio* $q(x)$ *de grau* $m-1$ *tal que*

$$h(x) = (x-\alpha)q(x) + h(\alpha).$$

Vamos usar este lema e indução no grau n do polinômio $f(x)$ para provar o teorema. Se $n = 1$, temos que $f(x) = x + b$, e a única solução possível em $\mathbb{Z}_p$ é $\overline{-b}$. Logo um polinômio de grau 1 tem exatamente uma raiz em $\mathbb{Z}_p$. Neste caso provamos até mais do que pedia o teorema.

Suponha agora que todo polinômio de grau $k-1$, cujo termo líder é 1, tem no máximo $k-1$ raízes distintas em $\mathbb{Z}_p$. Esta é a hipótese de indução. Mostraremos, a partir disto, que a mesma afirmação vale para um polinômio de grau k, cujo termo líder é 1.

Se, por acaso, $f(x)$ não tiver nenhuma raiz em $\mathbb{Z}_p$, então o teorema é válido, já que $k \geqslant 0$. Isto pode acontecer; há um exemplo ao fim da demonstração. Digamos então que $\overline{\alpha}$ é uma raiz de $f(x)$ em $\mathbb{Z}_p$; isto é $f(\alpha) \equiv 0 \pmod{p}$. Pelo lema

$$f(x) = (x-\alpha)q(x) + f(\alpha), \tag{4.1}$$

onde $q(x)$ tem grau $k-1$. Observe que, como $f(x)$ e $x-\alpha$ têm coeficiente líder 1, isto também acontece com $q(x)$. Portanto podemos aplicar a hipótese de indução a $q(x)$. Reduzindo a equação (4.1) módulo p, obtemos

$$f(x) \equiv (x-\alpha)q(x) \pmod{p}. \tag{4.2}$$

Digamos que $\overline{\beta}$ é uma raiz de $f(x)$ em $\mathbb{Z}_p$, diferente de $\overline{\alpha}$. Isto significa que

$$f(\beta) \equiv 0 \pmod{p} \quad \text{mas} \quad \alpha - \beta \not\equiv 0 \pmod{p}.$$

Substituindo x por β em (4.2) e usando estas congruências

$$0 \equiv f(\beta) \equiv (\beta - \alpha)q(\beta) \pmod{p}.$$

Como p é primo isto implica que $q(\beta) \equiv 0 \pmod{p}$. Concluímos que se $\overline{\beta}$ é uma raiz de $f(x)$ em $\mathbb{Z}_p$ distinta de $\overline{\alpha}$, então $\overline{\beta}$ é raiz de $q(x)$ em $\mathbb{Z}_p$. Em outras palavras, $f(x)$ só pode ter uma raiz em $\mathbb{Z}_p$ além daquelas que também são raízes de $q(x)$. Mas, pela hipótese de indução, $q(x)$ tem no máximo $k-1$ raízes distintas em $\mathbb{Z}_p$. Portanto $f(x)$ não pode ter mais de k raízes distintas; o que conclui a demonstração.

Vamos considerar alguns exemplos. O primeiro é o polinômio $f(x) = x^2 + 3$. Este polinômio se encaixa nas condições do teorema, mas não tem raízes módulo 5. De fato, o quadrado de qualquer número módulo 5 é congruente a 1 ou 4. Este é o exemplo mencionado na demonstração do teorema.

O segundo exemplo ilustra o que acontece quando tentamos achar as raízes de uma equação sob um módulo que é composto. Por exemplo, as raízes da equação $x^2 - 170$ em $\mathbb{Z}_{385}$ são $\overline{95}$, $\overline{150}$, $\overline{235}$ e $\overline{290}$, como é possível constatar fazendo uma simples substituição. Temos assim uma equação de grau 2 com 4 raízes! Isto não contradiz o teorema, porque 385 é um número composto.

Para completar a demonstração do teorema falta-nos ainda provar o lema. Isso será feito, mais uma vez, usando indução. Entretanto, não conseguiríamos provar o lema usando o princípio de indução na forma em que foi enunciado na seção 2. Quando você ler a demonstração, vai descobrir porque.

Na forma como o princípio foi enunciado na seção 2, a hipótese de indução diz apenas que $P(k)$ tem que ser verdadeira. Na prática, se estamos tentando provar $P(k+1)$, já sabemos que $P(1), P(2), \ldots, P(k)$ são todas verdadeiras. Portanto essencialmente nada muda se assumirmos esta hipótese de indução mais forte. Para deixar clara a diferença, enunciaremos o princípio de indução novamente, mudando apenas a hipótese de indução.

Princípio de indução finita. *Seja* $P(n)$ *a proposição que queremos provar. Para que* $P(n)$ *seja verdadeira para todo* n *natural, basta que:*
(1) $P(1)$ *seja verdadeira;*
(2) Se $P(1), \ldots, P(k)$ *forem verdadeiras, então* $P(k+1)$ *também seja.*

Com isto fica fácil provar o lema por indução no grau m de $h(x)$. Se $m = 1$, existem inteiros a e b tais que $h(x) = ax + b$. Logo

$$h(x) = ax + b = a(x - \alpha) + a\alpha + b = a(x - \alpha) + h(\alpha).$$

Suponhamos, então, que o resultado vale para qualquer polinômio com coeficientes inteiros de grau $\leqslant m-1$. Disto queremos concluir que vale para $h(x)$, que tem grau m. Digamos que

$$h(x) = a_m x^m + a_{m-1}x^{m-1} + \cdots + a_1 x + a_0.$$

Seja

$$g(x) = h(x) - a_m x^{m-1}(x-\alpha),$$

isto é

$$g(x) = (a_{m-1} - a_m\alpha)x^{m-1} + a_{m-2}\,x^{m-2} + \cdots + a_1 x + a_0.$$

Logo $g(x)$ certamente tem grau $\leqslant m-1$. Mas nada nos garante que $g(x)$ tem grau exatamente $m-1$, porque poderia acontecer que $a_{m-1} - a_m\alpha$ fosse zero. É isto que nos impede de aplicar o princípio de indução na forma com que foi enunciado na seção 2. Observe ainda que $g(\alpha) = h(\alpha)$.

Pela hipótese de indução, como $g(x)$ tem grau $\leqslant m-1$, existe um polinômio $j(x)$ de coeficientes inteiros e grau menor ou igual a $m-2$ tal que

$$g(x) = j(x)(x-\alpha) + g(\alpha).$$

Como $g(\alpha) = h(\alpha)$,

$$g(x) = j(x)(x-\alpha) + h(\alpha).$$

Por outro lado,

$$h(x) = g(x) + a_m x^{m-1}(x-\alpha),$$

donde

$$h(x) = (j(x) + a_m x^{m-1})(x-\alpha) + h(\alpha).$$

Observe que $j(x) + a_m x^{m-1}$ tem grau $m-1$, pois $j(x)$ tem grau $m-2$. Isto completa a demonstração do lema.

Há uma outra maneira de provar o lema, muito mais imediata, mas que requer algum conhecimento básico de divisão de polinômios. Seja, novamente, $h(x)$ um polinômio de grau m e coeficientes inteiros. Dividindo $h(x)$ por $x-\alpha$ achamos dois polinômios $q(x)$ e $r(x)$ tais que

$$h(x) = q(x)(x-\alpha) + r(x), \tag{4.3}$$

e $r(x) = 0$ ou $r(x)$ tem grau menor que $x-\alpha$, que é o divisor. Logo $r(x) = c$ é uma *constante*, que pode ser nula ou não. Substituindo x por α na equação (4.3)

$$h(\alpha) = q(\alpha)(\alpha-\alpha) + c = c.$$

Portanto podemos reescrever (4.3) na forma

$$h(x) = q(x)(x - \alpha) + h(\alpha),$$

e acabou a demonstração.

5. Exercícios

1. Prove por indução que:
(1) $n^3 + 2n$ é divisível por 3 para todo inteiro $n \geqslant 1$.
(2) Se n é um inteiro positivo ímpar então $n^3 - n$ é divisível por 24.
(3) Um polígono convexo de n lados tem exatamente $n(n-3)/2$ diagonais.
(4) Para $n \geqslant 1$ inteiro, $\sum_{k=1}^{n} k(k+1) = n(n+1)(n+2)/3$.

2. Chamamos de *hexagonais* os números definidos pela fórmula $h_n = 1 + 3n(n-1)$ para $n = 1, 2, \ldots$. O nome vem do fato de que estes números de pontos podem ser dispostos em hexágonos regulares concêntricos.
(1) Calcule a soma dos n primeiros números hexagonais quando $n = 1, 2, 3, 4$, e 5. Use estes dados numéricos para advinhar a fórmula da soma dos n primeiros números hexagonais.
(2) Prove a fórmula obtida em (1) usando o princípio de indução finita.

3. O princípio de indução é muito útil, mas precisamos usá-lo com o devido cuidado, para não cair em ciladas. Por exemplo: o que há de errado com a seguinte demonstração de que em qualquer conjunto finito de bolas coloridas, todas as bolas têm que ter a mesma cor? A demonstração é por indução no número de elementos do conjunto:
Se o conjunto tem uma única bola, então todas as suas bolas têm a mesma cor; isto certamente é verdade. Suponhamos que em qualquer conjunto de k *bolas coloridas, todas as bolas têm a mesma cor. Queremos usar isto para mostrar que em um conjunto* S *de* $k+1$ *bolas coloridas (que vamos denotar por* $b_1, \ldots b_{k+1}$*), todas as bolas têm a mesma cor. Para provar isto, faça o seguinte. Do conjunto de* $k+1$ *bolas retire a bola* b_{k+1}*. Assim ficamos com as* k *bolas* $b_1, \ldots b_k$ *que, pela hipótese de indução, terão todas a mesma cor. Para concluir a demonstração, basta mostrar que* b_{k+1} *tem a mesma cor das demais bolas. Ponha, então, a bola retirada de volta no conjunto, e retire uma outra bola, digamos* b_1*. Novamente temos um conjunto de* k *bolas, que desta vez é* $b_2, \ldots, b_{k+1}$*. Pela hipótese de indução todas estas bolas devem ter a mesma cor. Logo* b_2 *e* b_{k+1} *têm a mesma cor. Portanto todas as bolas de* S *têm a mesma cor, já que* $b_1, \ldots, b_k$ *são todas bolas de uma mesma cor!*

4. São dadas 3^n moedas de real, uma das quais foi adulterada e pesa menos

do que devia. Você tem uma balança de dois pratos mas não tem pesos. A única forma de pesagem permitida consiste em pôr algumas moedas em cada prato e verificar se a balança está equilibrada. Mostre, por indução finita, que n pesagens deste tipo são suficientes para achar a moeda adulterada.

5. Vamos denotar o n-ésimo primo por p_n. Assim $p_1 = 2$, $p_2 = 3$ e $p_3 = 5$, por exemplo. O objetivo do exercício é achar uma cota superior para o n-ésimo primo em função de n.
(1) Mostre que $p_{n+1} \leqslant p_1 \ldots p_n + 1$.
(2) Use indução finita e (1) para mostrar que o n-ésimo número primo satisfaz a desigualdade $p_n \leqslant 2^{2^n}$.

6. Mostre, usando o Teorema de Fermat, que $2^{70} + 3^{70}$ é divisível por 13.

7. Seja a um número inteiro positivo escrito em base 10. Mostre que os algarismos das unidades de a^5 e a são os mesmos.

8. Use o Teorema de Fermat para provar que para todo inteiro n, o número $n^3 + (n+1)^3 + (n+2)^3$ é divisível por 9.

9. Seja p um número primo diferente de 2 e 5. Mostre que p divide um dos números do conjunto:

$$\{1, 11, 111, 1111, 11111, \ldots\}.$$

Sugestão: pelo Pequeno Teorema de Fermat, temos que se $p > 5$ então $10^{p-1} - 1$ é divisível por p. O caso $p = 3$ tem que ser tratado separadamente.

10. Mostre que a equação $x^{13} + 12x + 13y^6 = 1$ não admite soluções inteiras.

Sugestão: Reduza módulo 13 e use Fermat.

11. Calcule o resto da divisão de:
(1) $39^{50!}$ por 2251;
(2) 19^{39^4} por 191.

12. O objetivo desta questão é mostrar que se $p = 4n + 1$ é primo, então existem inteiros a e b tais que p divide $a^2 + b^2$. Sejam x e y dois números inteiros primos com p. Ponha $a = x^n$ e $b = y^n$. Então

$$(a^2 - b^2)(a^2 + b^2) = x^{4n} - y^{4n}.$$

(1) Use o teorema de Fermat para mostrar que $x^{4n} - y^{4n}$ é divisível por p.
(2) Use (1) para mostrar que p divide $a^2 + b^2$ ou $a^2 - b^2$.

Se p dividir a^2+b^2 o resultado está provado. Suponha, então, por absurdo que, para quaisquer inteiros x e y, temos que p divide $x^{2n}-y^{2n}=a^2-b^2$. Em particular isto tem que valer quando x é um inteiro qualquer e $y=1$. Isto nos diz que $x^{2n}-1$ é divisível por p qualquer que seja x inteiro.

(3) Mostre que se $x^{2n} \equiv 1 \pmod{p}$ para todo inteiro x, temos uma contradição com o teorema provado na seção 4.
(4) Reúna tudo isto para concluir que existem inteiros a e b tais que a^2+b^2 é divisível por p.

Fermat sabia que, na verdade, qualquer primo da forma $4n+1$ pode ser escrito como soma dos quadrados de dois inteiros. Sobre isto veja [**Weil 1987, Cap. II, §§VII e VIII**] ou [**Hefez 1993, Cap. 10, seção 2**]. Compare o resultado acima com o do exercício 10 do capítulo 4.

13. O objetivo desta questão é dar uma demonstração do teorema de Fermat, devida a L. Euler, e que não usa indução. Seja p um primo e $\overline{a} \neq \overline{0}$ um elemento de $U(p)$. Considere o subconjunto

$$S = \{\overline{a}, \overline{2a}, \overline{3a}, \ldots, \overline{(p-1)a}\}$$

de $U(p)$.
(1) Mostre que os elementos de S são todos *distintos*.
(2) Mostre que S tem $p-1$ elementos e conclua que $S = U(p)$.
(3) Mostre que (2) implica que o produto de todos os elementos de S é igual a $\overline{(p-1)!} = \overline{1} \cdot \overline{2} \cdot \cdots \cdot \overline{p-1}$.
(4) Mas o produto dos elementos de S também é igual a $\overline{a}^{(p-1)}\overline{(p-1)!}$. Iguale isto ao valor obtido em (3) e cancele o termo comum. O resultado é o teorema de Fermat.

14. Seja p um número primo e a um inteiro que não é divisível por p. Mostre que o inverso de $\overline{a}$ em $\mathbb{Z}_p$ é $\overline{a}^{p-2}$.

15. Seja $p = 4k+3$ um primo positivo. Dado $a \in \mathbb{Z}$, considere a equação $x^2 \equiv a \pmod{p}$.
(1) Dê exemplos de valores de a e p para os quais a equação *não* tem solução.
(2) Mostre que se a equação tem solução, então as únicas soluções módulo p são $\pm a^{k+1}$.

Sugestão: Se a equação tem solução, então existe um inteiro b tal que $b^2 \equiv a \pmod{p}$. Assim temos que

$$(a^{k+1})^2 \equiv (b^{4(k+1)}) \equiv b^{4k+2} \cdot b^2 \pmod{p}.$$

O resultado enunciado em (2) segue, agora, pelo teorema de Fermat.

16. Use o exercício 14 para escrever um programa que, tendo como entrada a e p, calcula o inverso de a módulo p. Por segurança faça seu programa verificar primeiro que p realmente *não* divide a. Use isto para escrever um outro programa que determine a solução de $ax \equiv b \pmod{p}$, dadas as entradas a, b e p.

17. Seja $p = 4k+3$ um primo positivo. Escreva um programa que, tendo por entrada p e um inteiro positivo a, calcula as duas soluções de $x^2 \equiv a \pmod{p}$. Observe que sabemos do exercício 15 que *se esta equação tem solução* b então $b \equiv a^{k+1} \pmod{p}$. Assim o programa deve calcular a forma reduzida de a^{k+1} módulo p e, em seguida, verificar se, de fato, esta é uma solução da equação dada. A saída será constituída pelas soluções da equação, ou por uma mensagem indicando que a equação dada não tem solução. Este exercício é o segundo de uma seqüência que termina com o exercício 8 do capítulo 11.

18. Vários problemas da teoria de números (como o último teorema de Fermat) requerem primos p que satisfaçam a congruência

$$a^{p-1} \equiv 1 \pmod{p^2}$$

para algum inteiro a. Escreva um programa que, tendo por entrada dois inteiros positivos $a > 1$ e r, determina os primos p que satisfazem a congruência acima e tais que $a + 1 \leqslant p \leqslant r$. Observe que o programa terá que aplicar o crivo de Eratóstenes para achar os primos menores ou iguais a r. Estes primos serão então testados na congruência acima. Se $r = 10^5$ então o número de primos satisfazendo a congruência é dois quando $a = 2, 5, 10$ e 14; e cinco quando $a = 19$, por exemplo.

6

Pseudoprimos

É hora de tratarmos de algumas aplicações mais substanciais do teorema de Fermat. Neste capítulo veremos como usá-lo para identificar que um número é composto *sem fatorá-lo*.

1. Pseudoprimos

De acordo com o teorema de Fermat, se p é primo e a é um inteiro qualquer, então $a^p \equiv a \pmod{p}$. Imagine então a seguinte situação. Temos um número ímpar n, e desejamos saber se é, ou não, composto. Digamos que descobrimos um inteiro b tal que $b^n \not\equiv b \pmod{n}$. Pergunta-se: n pode ser primo? Claro que não, porque a inequação acima viola o teorema de Fermat. Isto nos dá uma maneira indireta de verificar se um número é composto. Observe que, na prática, só precisamos considerar os inteiros b no intervalo $1 < b < n-1$: por quê? Em primeiro lugar, como estamos trabalhando módulo n, qualquer inteiro é congruente a um inteiro no intervalo de 0 a $n-1$. Além disso, a equação $b^n \equiv b \pmod{n}$ é *sempre* satisfeita quando b é 0, 1 ou $n-1$.

Este teste produz uma situação surpreendente. Com ele podemos chegar à conclusão de que um número é composto mesmo que não nos seja possível determinar seus fatores. Antes de fazer um exemplo é conveniente formular o teste de uma maneira mais fácil de utilizar na prática. Para isto usaremos a segunda versão do teorema de Fermat enunciada na seção 3 do capítulo anterior.

Teste. *Se $n > 0$ e $1 < b < n-1$ são números inteiros e $b^{n-1} \not\equiv 1 \pmod{n}$, então n é um número composto. O número b é conhecido como uma testemunha do fato de n ser composto.*

Vamos fazer um exemplo. Lembre-se que definimos $R(n)$ como sendo o número formado por n algarismos 1; ver exercício 5 do capítulo 2. Considere,

$R(229)$: queremos saber se é um número primo. Não é fácil achar um fator, mas usando um sistema de computação algébrica, vemos que $2^{R(229)-1}$ tem resto igual a

$$1045165005843333977817537688859828354886127372338848985708482884$$
$$0566689840629082553655231345237426825653914552760612156751288528$$
$$7283062854774198632697829520351103663852079821692412346101479040$$
$$743884170069248576365931104545032921 7.$$

que não é congruente a 1 módulo $R(229)$. Portanto $R(229)$ é composto.

Será que podemos inverter isto e usar o teorema de Fermat para verificar se um número é primo? Na verdade, estamos perguntando se um número ímpar n que satisfaz $b^{n-1} \equiv 1 \pmod{n}$, para *algum* $1 < b < n-1$, é primo? Leibniz, famoso pela invenção do cálculo (quase ao mesmo tempo que Newton), achava que sim, e usou isto como um critério de primalidade. Na verdade, ele tomava $b = 2$, que é o caso mais simples de calcular.

Infelizmente isto *não* é verdade. Por exemplo, $2^{340} \equiv 1 \pmod{341}$. Logo, segundo Leibniz, 341 seria um número primo. Mas $341 = 11 \cdot 31$, é composto. Estes 'falsos primos' são conhecidos como pseudoprimos. Isto é, um inteiro positivo n, ímpar e *composto*, é um *pseudoprimo* para a *base* b (onde $1 < b < n-1$) se $b^{n-1} \equiv 1 \pmod{n}$. Assim 341 é um pseudoprimo para a base 2.

Apesar de às vezes dar errado, o teste de Leibniz é muito útil. Pelo menos para números pequenos, ele acerta mais do que erra. Por exemplo, entre 1 e 10^9 existem 50.847.534 primos, mas apenas 5597 pseudoprimos para a base 2. Logo um número menor que um milhão que passa no teste de Leibniz tem uma alta probabilidade de ser primo.

Além disto, usamos apenas uma base até agora. Por que não testar para mais de uma base? Fazendo isto o teste fica ainda mais eficiente. Por exemplo, $3^{340} \equiv 56 \pmod{341}$. Logo 3 é uma testemunha de que 341 é composto. Na verdade, há apenas 1272 pseudoprimos para as bases 2 e 3 entre 1 e 10^9.

Como estamos limitando nossas bases ao intervalo 1 a $n-1$, poderíamos considerar a possibilidade de testar se n é pseudoprimo para todas as bases. Se n é grande, isto vai ser impossível, logo o interesse prático é muito limitado. Por outro lado, este problema leva a algumas questões surpreendentes, e de real interesse prático, como veremos mais adiante.

2. Números de Carmichael

Não existem números que sejam pseudoprimos para todas as bases. Isto é fácil de constatar. Se n é composto, então tem um fator b. Como

$$\text{mdc}(b^{n-1}, n) \neq 1,$$

temos pelo teorema de inversão que b não pode ser inversível módulo n. Em particular, $b^{n-1} \not\equiv 1 \pmod{n}$. Logo n não é pseudoprimo para a base b. Entretanto, pode ocorrer que um número composto n seja pseudoprimo para todas as bases que são primas a n. É este tipo de número que desejamos discutir nesta seção.

É preferível reformular o problema da seguinte maneira. Seja n um número inteiro positivo. Como caracterizar os números compostos n que satisfazem a equação $b^n \equiv b \pmod{n}$? A vantagem desta formulação é que podemos dispensar a condição $\text{mdc}(b, n) = 1$, o que é muito conveniente. O primeiro matemático a dar exemplos destes números foi R.D. Carmichael, em um artigo publicado em 1912; veja [**Carmichael 1912**]. Por isso são chamados números de Carmichael.

Portanto, um número composto ímpar $n > 0$ é um *número de Carmichael* se $b^n \equiv b \pmod{n}$ para todo $1 < b < n - 1$. Observe que, para um número ser de Carmichael é preciso que seja *composto*. Um número primo p também satisfaz à equação $b^p \equiv b \pmod{p}$, mas *não* é número de Carmichael.

Como o próprio Carmichael determinou, o menor número de Carmichael é 561. Em princípio podemos verificar isto usando a definição. Entretanto, mesmo para um número relativamente pequeno como este, isto é difícil de fazer usando apenas lápis e papel. Afinal, para mostrar que 561 é de Carmichael *pela definição*, precisamos verificar que $b^{561} \equiv b \pmod{561}$ para $b = 2, 3, 4, \ldots, 559$; o que dá um total de 558 bases a serem testadas, algumas não tão pequenas. Não há dúvida de que esta é uma tarefa para um computador. Entretanto, mesmo um computador razoável pode ter dificuldade em usar este método para verificar que

$$3494075153422874350506032047195872 01$$

é um número de Carmichael.

Felizmente há uma maneira mais simples de verificar que um dado número composto é de Carmichael. Consideremos 561 mais uma vez. Podemos facilmente fatorá-lo

$$561 = 3 \cdot 11 \cdot 17.$$

Seja agora b um número inteiro entre 2 e 559. Queremos mostrar que

$$b^{561} \equiv b \pmod{561}. \tag{2.1}$$

Nossa estratégia será a seguinte: mostraremos que $b^{561} - b$ é divisível por 3, por 11 e por 17. Como estes são primos *distintos*, segue do lema da seção 6 do capítulo 2 que o produto destes primos divide $b^{561} - b$. Mas este produto é 561, e dizer que 561 divide $b^{561} - b$ é equivalente a (2.1).

Só nos resta mostrar que $b^{561} - b$ é divisível por cada um dos fatores de 561, separadamente. Como estes fatores são primos, o teorema de Fermat vem em nosso socorro. Vamos efetuar as contas para o primo 17, e deixar 3 e 11 por sua conta. Queremos verificar que $b^{561} - b$ é divisível por 17; isto é, que

$$b^{561} \equiv b \pmod{17}. \tag{2.2}$$

Há dois casos a considerar. Se 17 divide b então ambos os membros de (2.2) são congruentes a zero módulo 17; logo a congruência é imediatamente verificada. Digamos, então, que 17 não divide b. Neste caso o teorema de Fermat nos diz que $b^{16} \equiv 1 \pmod{17}$. Para aplicar isto em (2.2) precisamos dividir 561 por 16; mas isto dá $561 = 35 \cdot 16 + 1$. Assim

$$b^{561} \equiv (b^{16})^{35} \cdot b \equiv b \pmod{17}$$

onde a segunda congruência segue do teorema de Fermat. Observe que só podemos aplicar o teorema de Fermat de uma maneira tão simples porque o resto da divisão de 561 por 16 deu 1. Por sorte o resto da divisão de 561 por $2(= 3 - 1)$ e por $10(= 11 - 1)$ também é 1. Por isto os cálculos para os primos 3 e 11 serão exatamente análogos ao caso 17.

Para que a estratégia acima pudesse ser posta em prática, precisamos de um pouco de sorte. Afinal, o resto da divisão de 561 por cada um de seus fatores primos menos um deu 1, o que nos ajudou a aplicar o teorema de Fermat. Mas houve um outro golpe de sorte, um pouco menos óbvio. Só pudemos aplicar o lema da seção 6 do capítulo 2 porque 561 é um produto de primos distintos–isto é, a multiplicidade de cada primo na fatoração de 561 é 1.

Estas considerações sugerem que não é provável que tenhamos tanta sorte para outros números de Carmichael. A não ser que haja muito poucos números de Carmichael. A verdade é que há *infinitos* números de Carmichael, e para todos eles as propriedades observadas no caso especial do 561 são satisfeitas. Curiosamente isto já havia sido observado por A. Korselt, 13 anos antes de Carmichael publicar seu artigo. Entretanto Korselt

não deu nenhum exemplo de um número satisfazendo estas propriedades. O resultado de Korselt é o conteúdo do seguinte teorema.

Teorema de Korselt. *Um inteiro positivo ímpar* n *é um número de Carmichael se, e somente se, cada fator primo* p *de* n *satisfaz as duas condições seguintes:*
(1) p^2 *não divide* n*;*
(2) $p-1$ *divide* $n-1$.

Vamos mostrar primeiro que se um número n satisfaz as condições (1) e (2) acima, então é um número de Carmichael. Faremos isto usando a mesma estratégia que empregamos para 561. Digamos que p é um fator primo de n. Começaremos mostrando que

$$b^n \equiv b \pmod{p}. \tag{2.3}$$

Se b é divisível por p, então ambos os membros de (2.3) são congruentes a zero; logo a congruência é imediatamente verificada. Digamos, então, que p não divide b. Neste caso o teorema de Fermat nos diz que $b^{p-1} \equiv 1 \pmod{p}$. Para aplicar isto em (2.3) precisamos dividir n por $p-1$. Mas a condição (2) nos diz que $p-1$ divide $n-1$; isto é $n-1=(p-1)q$, para algum inteiro positivo q. Portanto

$$n = (n-1)+1 = (p-1)q+1.$$

Assim

$$b^n \equiv (b^{p-1})^q \cdot b \equiv b \pmod{p}$$

onde a segunda congruência segue do teorema de Fermat. Resumindo: se p é um fator primo de n então $b^n \equiv b \pmod{p}$ para qualquer inteiro b.

Voltando às hipóteses do teorema, temos de (1), que $n = p_1 \ldots p_k$ onde $p_1 < \cdots < p_k$ são primos distintos. Mas já vimos que $b^n - b$ é divisível por cada um desses primos. Como são primos distintos, podemos concluir pelo lema da seção 6 do capítulo 2, que $b^n - b$ é divisível pelo produto $p_1 \ldots p_k = n$. Em outras palavras $b^n \equiv b \pmod{n}$. Como estes cálculos valem para qualquer inteiro b, temos que n é um número de Carmichael.

Mostramos, assim, que se n satisfaz as condições (1) e (2) então é um número de Carmichael. Passemos à recíproca. Em primeiro lugar desejamos verificar que se n é um número de Carmichael então a condição (1) é satisfeita. Provaremos a contrapositiva disto: se existe um primo p para o qual p^2 divide n, então n não pode ser um número de Carmichael. Suponhamos, então, que p é um primo e que p^2 divide n. Para mostrar que n

não é número de Carmichael, basta achar um inteiro b para o qual $b^n \not\equiv b \pmod{n}$. Escolheremos $b = p$. Temos que

$$p^n - p = p(p^{n-1} - 1).$$

Mas p não divide $p^{n-1} - 1$, logo p^2 não pode dividir $p^n - p$. Como estamos supondo que p^2 divide n, isto significa que n não pode dividir $p^n - p$. Em outras palavras $p^n \not\equiv p \pmod{n}$, e n não é número de Carmichael.

Para completar a demonstração, falta-nos apenas mostrar que se n é um número de Carmichael, então a condição (2) é satisfeita. Entretanto isto depende do *teorema da raiz primitiva* que só será provado no capítulo 10. Por isso o final da demonstração do teorema de Korselt só será feito na seção 3 do capítulo 10.

Para verificar que um número é de Carmichael usando a caracterização dada pelo teorema de Korselt basta que possamos fatorá-lo. Infelizmente esta pode não ser uma tarefa muito fácil, como já sabemos. Por sorte há muitos números de Carmichael grandes cujos fatores são bem pequenos. Por exemplo, o número de 36 algarismos dado no começo da seção é o menor número de Carmichael com 20 fatores primos; a fatoração completa deste número é

$$11 \cdot 13 \cdot 17 \cdot 19 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 61 \cdot 71 \cdot 73 \cdot 97 \cdot 101 \cdot 109 \cdot 113 \cdot 151 \cdot 181 \cdot 193 \cdot 641.$$

Usando um sistema de computação algébrica é muito fácil verificar que este é um número de Carmichael. Para uma família que produz vários exemplos de números de Carmichael, veja o exercício 3.

Em seu artigo de 1912, depois de dar exemplo de 15 dos números que hoje levam seu nome, Carmichael acrescentou *'esta lista pode ser estendida indefinidamente'*. Contudo, logo tornou-se claro que seria difícil provar isto. Surgia assim o problema de mostrar que há infinitos números de Carmichael. Não é surpreendente que o problema seja difícil, já que os números de Carmichael são bastante raros. Por exemplo, entre 1 e 10^9 há apenas 646 destes números, contra 50.847.534 primos. Esta questão foi finalmente resolvida em um artigo publicado em 1994. Inspirando-se em idéias de P. Erdös, os matemáticos Alford, Granville e Pomerance provaram que existem infinitos números de Carmichael; veja [**Alford, Granville e Pomerance 1994**]. Um outro resultado publicado neste artigo tem grande importância prática para os testes de primalidade usados por sistemas de computação algébrica e será discutido na próxima seção.

3. Teste de Miller

Até agora vimos que o teorema de Fermat dá lugar a um teste que detecta números compostos com uma eficiência razoável, mas que não é bom se o que desejamos é verificar se um número é primo. A existência dos números de Carmichael significa que, mesmo aplicando o teste para um número relativamente grande de bases, não podemos ter muita segurança quanto à primalidade do número testado. Entretanto podemos melhorar muito este teste, com uma modificação bastante simples. Este novo teste foi introduzido em 1976 por G.L. Miller.

O *teste de Miller* tem seu ponto de partida na seguinte observação. Seja $n > 0$ um número *ímpar* e $1 < b < n-1$ um inteiro. Como n é ímpar, temos que $n-1$ é par. Digamos que $n-1 = 2^k q$, onde q é um inteiro ímpar e $k \geqslant 1$. Isto é, fatoramos a maior potência de 2 possível em $n-1$.

O teste consiste em calcular a seguinte seqüência de potências *módulo* n:

$$b^q,\ b^{2q}, \ldots, b^{2^{k-1}q}, b^{2^k q}.$$

Se n for um número primo, então pelo menos uma destas potências tem que ser congruente a 1 módulo n. De fato, pelo teorema de Fermat

$$b^{2^k q} \equiv b^{n-1} \equiv 1 \pmod{n}.$$

Digamos que j é o *menor* expoente tal que $b^{2^j q} \equiv 1 \pmod{n}$. Se $j \geqslant 1$, podemos usar produtos notáveis para escrever

$$b^{2^j q} - 1 = (b^{2^{j-1}q} - 1)(b^{2^{j-1}q} + 1).$$

Como estamos supondo que n é primo e divide $b^{2^j q} - 1$, então ou n divide $b^{2^{j-1}q} - 1$ ou n divide $b^{2^{j-1}q} + 1$. Mas j é o *menor* expoente tal que $b^{2^j q} - 1$ é divisível por n. Portanto, $b^{2^{j-1}q} - 1$ não pode ter n como fator. Resta-nos que $b^{2^{j-1}q} + 1$ seja divisível por n; isto é $b^{2^{j-1}q} \equiv -1 \pmod{n}$.

Concluímos dos cálculos acima que se n é primo, então uma das potências da seqüência

$$b^q, b^{2q}, \ldots b^{2^{k-1}q}.$$

tem que ser congruente a -1 módulo n. Bem, isto não é exatamente verdadeiro, porque as contas acima dependem do fato de que $j \geqslant 1$. Se $j = 0$ então $b^q \equiv 1 \pmod{n}$. Mas a esta congruência não podemos aplicar o produto notável, já que q é ímpar. Portanto, se n é primo então uma das potências da seqüência é congruente a -1 ou $b^q \equiv 1 \pmod{n}$. Se nada disto acontecer, então n é composto.

Mais uma vez temos um teste que, com um pouco de sorte, detecta números compostos. A vantagem deste teste sobre o anterior é que descobre números compostos mesmo entre os pseudoprimos. O algoritmo é o seguinte.

Teste de Miller.
Entrada: *um inteiro ímpar* n, *que se quer testar, e a base* b, *onde* $1 < b < n-1$.
Saída: *uma das mensagens: '*n *é composto' ou 'teste inconclusivo'.*

Etapa 1: *Divida* $n-1$ *sucessivamente por* 2 *até encontrar* q *(um número ímpar) e* k *tais que* $n-1 = 2^k q$.
Etapa 2: *Comece fazendo* $i = 0$ *e* $r =$ *resto de* b^q *por* n.
Etapa 3: *Se* $i = 0$ *e* $r = 1$ *ou se* $i \geqslant 0$ *e* $r = n-1$ *a saída é 'teste inconclusivo'.*
Etapa 4: *Incremente* i *de* 1 *e substitua* r *pelo resto da divisão de* r^2 *por* n.
Etapa 5: *Se* $i < k$ *volte à Etapa 3, senão a saída é '*n *é composto'.*

Em princípio, quando o teste tem saída *inconclusivo*, duas coisas podem acontecer: n pode ser primo, ou n pode ser composto. Infelizmente a segunda hipótese acontece na prática. Vejamos alguns exemplos, começando com uma situação conclusiva. Já vimos que o teste da seção 1 não detecta que 341 é primo usando apenas 2 como base. Para aplicar o teste de Miller, começamos verificando que $340 = 2^2 \cdot 85$. Precisamos calcular as potências de 2 módulo 341 para os expoentes 85 e 170, temos que

$$2^{85} \equiv 32 \pmod{341}$$
$$2^{170} \equiv 32^2 \equiv 1 \pmod{341}.$$

Portanto 341 é composto pelo teste de Miller.

Outro exemplo interessante é o número de Carmichael 561. O teste da seção 1 é praticamente inútil para este tipo de número. Vamos ver o que acontece quando aplicamos o teste de Miller para a base 2. Um cálculo rápido mostra que $560 = 2^4 \cdot 35$. A seqüência de restos módulo 561 das potências de 2 é

expoentes	35	$2 \cdot 35$	$2^2 \cdot 35$	$2^3 \cdot 35$
restos	263	166	67	1

Portanto 561 tem que ser composto. Note que chegamos a esta conclusão testando somente para a menor base possível.

Vamos agora aos exemplos ruins. O mais simples é 25. Vejamos o que acontece se escolhermos 7 como base. Como $24 = 2^3 \cdot 3$, a seqüência de restos

é

expoentes	3	$2 \cdot 3$	$2^2 \cdot 3$
restos	18	24	1

Observe que $24 \equiv -1 \pmod{25}$, portanto o teste é inconclusivo, embora 25 seja um número composto. É claro que a escolha de 7 como base foi proposital (e capciosa). É fácil verificar que, aplicando o teste de Miller a 25 para a base 2, temos como saída que o número é composto.

Se um número *composto* n tem resultado inconclusivo para o teste de Miller com respeito a uma base b, dizemos que n é um *pseudoprimo forte* para a base b. No exemplo acima, 25 é um pseudoprimo forte para a base 7. É importante observar que *todo pseudoprimo forte para uma certa base é pseudoprimo para aquela base*. Isto requer uma demonstração, que é apenas uma conta fácil, e fica como exercício.

Como já vimos, 25 *não* é um pseudoprimo forte para a base 2. O menor pseudoprimo forte para a base 2 é 2047. E mais, há apenas 1282 pseudoprimos fortes entre 1 e 10^9. Isto já dá uma boa medida da eficiência deste teste. É claro que, como no caso do teste da seção 1, podemos aplicar o teste de Miller para várias bases, o que o torna ainda mais eficiente. Na verdade, se um inteiro n é submetido ao teste de Miller para mais de $n/4$ bases entre 1 e $n-1$, e tem por saída inconclusivo, então n tem que ser primo; veja [**Lemos 1989, Cap. 3, seção 2, proposição 2.1**]. Isto é um fato interessante, mas é pouco útil na prática. Se n tem 200 algarismos, não há a menor chance de aplicar o teste a tantas bases e ainda estar vivo quando o resultado for obtido.

4. Primalidade e computação algébrica

Na vida real o que se faz é aplicar o teste de Miller para uma quantidade de bases que se acredita ser 'suficientemente grande'. É assim que muitos sistemas de computação algébrica verificam se um número é primo. Por exemplo, o *ScratchPad*, um sistema desenvolvido pela IBM, afirma que um número é primo se tem como saída inconclusivo para o teste de Miller quando aplicado a 10 bases previamente escolhidas. As bases usadas são os dez primeiros números primos. É claro que isto não é uma garantia de primalidade. Por exemplo, existe um número (de 337 algarismos) que é um pseudoprimo forte para todas as bases primas até 200. Neste caso estamos testando não 10, mas 46 bases!

Cada sistema de computação algébrica tem sua própria estratégia. Por exemplo, o *Maple V.2* testa primalidade em três etapas. Primeiramente é

feita uma busca para ver se o número tem algum fator menor que mil. Se não tiver, então é aplicado o teste de Miller para as bases $2, 3, 5, 7$ e 11. Se o usuário desejar, é possível inclusive escolher um número maior de bases. Finalmente o sistema verifica que o número dado não é da forma

$$(u+1)\left(k\frac{u}{2}+1\right) \quad \text{para} \quad 3 \leqslant k \leqslant 9,$$

nem

$$(u+1)(ku+1) \quad \text{para} \quad 5 \leqslant k \leqslant 20.$$

A justificativa para este último estágio é que há muitos exemplos de números destas duas formas que são pseudoprimos fortes para as bases adotadas como padrão pelo sistema. Infelizmente o número

$$1253075960778449601058457392 3$$

passa o teste do *Maple V.2* mas é composto: um de seus fatores primos é 286472803.

O *Axiom 1.1*, o sistema da IBM que sucedeu ao *ScratchPad*, ajusta a escolha das bases de acordo com o número a ser testado. O sistema também verifica se o número dado pertence a algumas famílias reconhecidamente problemáticas, de maneira semelhante ao *Maple V.2*. Pode-se mesmo provar que o teste do *Axiom 1.1* sempre detecta corretamente primos menores que 341550071728321. Mais uma vez, o teste não é perfeito e sucumbe a um número composto de 56 algarismos.

Mas a verdade completa é ainda pior. Uma das conseqüências do trabalho de Alford, Granville e Pomerance sobre números de Carmichael é a seguinte: *dado um número finito qualquer de bases, existem infinitos números de Carmichael que são pseudoprimos fortes para todas estas bases*. Este resultado é um alerta importante, e nos diz que é sempre extremamente perigoso afirmar a primalidade de um número com base no resultado do teste de Miller. Em vista deste resultado, a versão 2.2 do *Axiom* ajusta o número de bases testadas de acordo com o tamanho do número: são usadas k bases para um número de $2k$ algarismos. Isto aumenta muito a segurança do teste. Para mais detalhes consulte [**Arnault 1995**].

Tudo isto nos leva a concluir que os testes apresentados neste capítulo são bons para detectar que um número é composto ou para nos dar uma boa margem de probabilidade de que um número é primo. Mais adiante veremos testes que garantem que um número é primo; infelizmente estes testes são muito mais lentos e difíceis de usar.

5. Exercícios

1. Quais dos seguintes números são pseudoprimos para a base 2: 645, 567 e 701? Quais são pseudoprimos para a base 3? Quais são primos?

2. Mostre que se n é um pseudoprimo para as bases a e ab, então n também é um pseudoprimo para a base b.

3. Seja n um inteiro positivo. Ponha $p_1 = 6n+1$, $p_2 = 12n+1$ e $p_3 = 18n+1$. Mostre que se p_1, p_2 e p_3 são primos, então o produto $p_1p_2p_3$ é um número de Carmichael. Mostre que estas condições são satisfeitas se $n = 1$, 6 e 35. Quais os números de Carmichael obtidos para cada um destes valores de n?

4. Fatore 29341 e mostre que é um número de Carmichael.

5. Sejam $p_1 < p_2$ dois primos. Ponha $n = p_1p_2$ e digamos que $p_1 - 1$ e $p_2 - 1$ dividem $n-1$. Mostre que $n-1 \equiv p_1 - 1 \pmod{p_2 - 1}$ e obtenha uma contradição. Conclua que um número de Carmichael não pode ser o produto de apenas dois primos.

6. Quais dos seguintes números são pseudoprimos fortes para a base 2: 645, 2047 e 2309? Quais são pseudoprimos fortes para a base 3? Quais são primos?

7. Mostre que se um número n é pseudoprimo *forte* para a base b então é um pseudoprimo para esta mesma base.

8. Escreva um programa que, tendo como entrada um inteiro positivo r, determina todos os números entre 1 e r que são pseudoprimos para as bases 2 e 3. Lembre-se que um tal pseudoprimo n é um número *ímpar* e *composto* que satisfaz as equações

$$2^{n-1} \equiv 1 \pmod{n} \quad \text{e} \quad 3^{n-1} \equiv 1 \pmod{n}.$$

Portanto o programa só precisa testar os números ímpares e além destas congruências, tem que verificar se os números obtidos são primos ou não. Uma maneira de fazer isto é programar o Crivo de Eratóstenes de modo a conservar a lista dos ímpares compostos, em vez dos primos, e depois testar as congruências. Aplique o programa com $r = 10^5$. Quantos e quais foram os pseudoprimos obtidos? Quantos destes são números de Carmichael?

9. O objetivo deste exercício é a elaboração de um programa para determinar todos os números de Carmichael que são produtos de d primos, *todos menores que* 10^3. O principal problema é que, mesmo para valores relativamente

pequenos de d, alguns dos números de Carmichael obtidos podem ser bem grandes. Isto torna difícil verificar que $p-1$ divide $n-1$, quando p é um fator primo de n. Mas é fácil contornar este problema usando congruências. Digamos que o programa gerou o número $n = p_1p_2\ldots p_d$. Precisamos verificar se $n-1$ é ou não divisível por p_1-1, por exemplo. Para isto basta calcular a forma reduzida de n módulo p_1-1. Como n aparece fatorado, você pode multiplicar os fatores um de cada vez e ir reduzindo módulo p_1-1 a cada produto, de modo a manter o resultado pequeno. Como os fatores são menores que 10^3 isto vai ser muito rápido de executar. Aplique o seu programa para $d = 1, \ldots, 8$. Quais os números de Carmichael obtidos em cada caso? Não é necessário multiplicar os fatores para obter o número, basta que o programa liste os fatores.

10. Escreva um programa para determinar o menor pseudoprimo forte para uma dada base. Vai ser necessário implementar o teste de Miller, de modo que a entrada seja um inteiro positivo $b > 1$. O programa deve aplicar o teste de Miller na base b aos ímpares compostos, até achar o primeiro número para o qual o teste é inconclusivo. Uma maneira de fazer isto é programar o Crivo de Eratóstenes de maneira a conservar a lista dos ímpares compostos, em vez dos primos e depois testar as congruências. Aplique o teste para as bases 2, 3, 5 e 7. Quais os resultados obtidos?

11. O objetivo deste exercício é a elaboração de um programa para determinar pseudoprimos para a base 2 da forma p^2, onde p é um primo. O programa deve ter como entrada um número inteiro positivo r. Em seguida será usado o crivo de Eratóstenes para encontrar os primos $p \leqslant r$. Finalmente o programa vai testar quais os valores de p para os quais p^2 é um pseudoprimo para a base 2. Os cálculos ficam um pouco mais eficientes se você lembrar que

$$2^{p^2} \equiv (2^p)^p \pmod{p^2}.$$

Há apenas dois exemplos de pseudoprimos quadrados para a base 2 quando $r = 5 \cdot 10^4$. Quais são?

7

Sistemas de congruências

Neste capítulo vamos estudar a solução de sistemas de equações lineares, o chamado *algoritmo chinês do resto*. O capítulo se encerrará com uma aplicação deste algoritmo a um método de criptografia para partilhar senhas.

1. Equações lineares

Começamos considerando o caso de uma única equação linear

$$ax \equiv b \pmod{n} \tag{1.1}$$

onde n é um inteiro positivo. Já vimos que estas equações são muito fáceis de resolver quando $\overline{a}$ é inversível em $\mathbb{Z}_n$. Vale a pena recordar. Digamos que $\overline{\alpha}$ é o inverso de $\overline{a}$ em $\mathbb{Z}_n$. Multiplicando a equação (1.1) por α obtemos

$$\alpha(ax) \equiv \alpha b \pmod{n}.$$

Como $\overline{\alpha a} = \overline{1}$ em $\mathbb{Z}_n$,

$$x \equiv \alpha b \pmod{n},$$

que é a solução da equação. Em particular, se n é primo e $a \not\equiv 0 \pmod{n}$ então (1.1) sempre tem solução.

Se $\overline{a}$ *não* tem inverso em $\mathbb{Z}_n$, o problema se torna mais complicado. Já vimos que $\overline{a}$ não ter inverso é equivalente a dizer que $\mathrm{mdc}(a, n) \neq 1$. Se a equação (1.1) tem solução, isto quer dizer que existem $x, y \in \mathbb{Z}$ tais que

$$ax - ny = b. \tag{1.2}$$

Mas isto só é possível se o $\mathrm{mdc}(a, n)$ divide b. Primeira conclusão: (1.1) *só tem solução quando* b *é divisível pelo* $\mathrm{mdc}(a, n)$. É claro que se $\overline{a}$ tem inverso em $\mathbb{Z}_n$ esta condição é satisfeita, porque neste caso $\mathrm{mdc}(a, n) = 1$.

Suponhamos então que $d = \mathrm{mdc}(a, n)$ divide b. Digamos que $a = da'$,

$b = db'$ e $n = dn'$. Substituindo em (1.2) e cancelando d,

$$a'x - n'y = b'.$$

Que se converte na equação $a'x \equiv b' \pmod{n'}$; uma nova equação em congruências. Mas cuidado, o módulo mudou. A equação original tinha n como módulo; a nova equação tem módulo n', um divisor de n. Observe que $\text{mdc}(a', n')$ é sempre 1, portanto esta última equação sempre tem solução. Parece tudo muito simples, mas esta nova equação guarda algumas surpresas.

Vejamos um exemplo. Seja $6x \equiv 4 \pmod 8$. Como $\text{mdc}(6, 8) = 2 \neq 1$, não podemos inverter $\overline{6}$ em $\mathbb{Z}_8$. Se esta equação tiver solução, então existem inteiros x e y tal que $6x - 8y = 4$. Dividindo toda a equação por 2, obtemos $3x - 4y = 2$. Isto nos leva a $3x \equiv 2 \pmod 4$. Mas $\overline{3}$ é seu próprio inverso em $\mathbb{Z}_4$. Multiplicando esta última equação por 3, chegamos à solução

$$x \equiv 2 \pmod 4. \tag{1.3}$$

Isto é um pouco insatisfatório. Começamos com uma equação módulo 8, mas obtivemos uma solução módulo 4. É fácil consertar isto, basta converter (1.3) em uma expressão em inteiros (isto é, sem congruências) e ver como as soluções se comportam módulo 8. Temos que (1.3) é equivalente a $x = 2 + 4k$, onde k é um inteiro qualquer. Há duas possibilidades para k. Se k for par, então $x \equiv 2 \pmod 8$ é uma das soluções. Se k for ímpar, então $k = 2m + 1$ e $x = 6 + 8m$. Isto é $x \equiv 6 \pmod 8$, que é uma outra solução. Portanto a equação $6x \equiv 4 \pmod 8$ tem duas soluções em $\mathbb{Z}_8$, que são $\overline{2}$ e $\overline{6}$. Observe que a equação é linear, mas tem duas soluções. Isto já não nos deve surpreender. Sabemos do Capítulo 5, seção 4, que o número de soluções distintas de uma equação de grau k é sempre $\leqslant k$ quando o módulo é primo. Mas vimos também que isto pode não acontecer quando o módulo é composto, que é o caso neste exemplo.

2. Um exemplo astronômico

O principal resultado deste capítulo será um algoritmo para resolver sistemas de congruências lineares. O algoritmo que estudaremos é muito antigo e foi inventado, independentemente, pelos chineses e pelos gregos, para resolver problemas de astronomia. Felizmente uma longa digressão sobre a astronomia chinesa de quase 2000 anos atrás não é pré-requisito para entendermos estas aplicações. Podemos formulá-las como problemas que fazem sentido em pleno fim do segundo milênio; por exemplo:

Três satélites passarão sobre o Rio esta noite. O primeiro à 1 hora da madrugada, o segundo às 4 horas e o terceiro às 8 horas da manhã. Cada satélite tem um período diferente. O primeiro leva 13 horas para completar uma volta em torno da Terra, o segundo 15 horas e o terceiro 19 horas. Determine quantas horas decorrerão, a partir da meia-noite, até que os três satélites passem ao mesmo tempo sobre o Rio.

Vejamos como formular o problema de maneira matemática.

Chamaremos de x o número de horas, contadas a partir da meia-noite de hoje, quando os três satélites passarão juntos sobre o Rio. O primeiro satélite passa sobre o Rio a cada 13 horas, a contar da 1 da madrugada. Logo precisamos ter que $x = 1 + 13t$, para algum inteiro positivo t. Equivalentemente $x \equiv 1 \pmod{13}$. As equações correspondentes aos outros dois satélites são

$$x \equiv 4 \pmod{15} \quad \text{e} \quad x \equiv 8 \pmod{19}.$$

Os três satélites passarão juntos sobre o Rio para os valores de x que satisfazem simultaneamente a estas três equações. Isto é, precisamos resolver o sistema

$$\begin{aligned} x &\equiv 1 \pmod{13} \\ x &\equiv 4 \pmod{15} \\ x &\equiv 8 \pmod{19} \end{aligned} \tag{2.1}$$

Observe que não podemos somar ou subtrair as equações entre si, já que os módulos são diferentes. Contornaremos este problema convertendo as equações de congruências, sempre que necessário, em identidades de inteiros.

Assim $x \equiv 1 \pmod{13}$ corresponde a $x = 1 + 13t$, que é um inteiro e, como tal, pode ser substituído na segunda equação, o que dá

$$1 + 13t \equiv 4 \pmod{15} \quad \text{ou seja} \quad 13t \equiv 3 \pmod{15}.$$

Mas 13 é inversível módulo 15 e seu inverso é 7. Multiplicando $13t \equiv 3 \pmod{15}$ por 7, e reduzindo os números módulo 15, obtemos

$$t \equiv 6 \pmod{15}.$$

Assim t é da forma $t = 6 + 15u$. Portanto

$$x = 1 + 13t = 1 + 13(6 + 15u) = 79 + 195u.$$

Observe que qualquer número da forma $79 + 195u$ satisfaz as duas primeiras congruências do sistema (2.1). Finalmente, vamos substituir $x = 79 + 195u$

na última equação do sistema. Com isto obtemos

$$79 + 195u \equiv 8 \pmod{19} \quad \text{logo} \quad 5u \equiv 5 \pmod{19}.$$

Como 5 é inversível módulo 19, podemos cancelá-lo da equação acima obtendo $u \equiv 1 \pmod{19}$. Reescrevendo esta equação em termos de inteiros, temos que $u = 1 + 19v$, para algum inteiro v. Substituindo isto em x

$$\begin{aligned} x &= 79 + 195u \\ &= 79 + 195(1 + 19v) \\ &= 274 + 3705v. \end{aligned}$$

O que podemos concluir a respeito dos satélites? Lembre-se que x corresponde ao tempo contado a partir da meia-noite. Logo queremos descobrir qual o menor valor positivo de x que satisfaz as equações; que é 274, claro. Portanto os satélites passarão juntos pela primeira vez 274 horas depois da meia-noite de hoje. Isto corresponde a 11 dias e 10 horas. Mas nossa solução nos diz mais. Somando um múltiplo qualquer de 3705 à 274 temos uma nova solução do sistema. Em outras palavras, os satélites voltarão a passar juntos a cada 3705 horas, que corresponde 154 dias e 9 horas.

Na próxima seção faremos uma análise detalhada do método acima. Observe que dividimos a solução deste sistema de 3 equações, de modo a resolver dois sistemas de duas equações. De fato, resolvendo as duas primeiras equações, obtivemos $x = 79 + 195u$. Isto corresponde a uma nova equação, que é $x \equiv 79 \pmod{195}$. Para obter o valor final de x, resolvemos o sistema

$$\begin{aligned} x &\equiv 79 \pmod{195} \\ x &\equiv 8 \pmod{19}. \end{aligned}$$

Em geral, a solução de um sistema de muitas equações é obtida através da solução de vários sistemas de duas equações. Por isso, na seção 3 é suficiente analisar o algoritmo correspondente à solução de um sistema de duas equações.

3. Algoritmo chinês do resto

O algoritmo chinês do resto tem este nome porque um dos primeiros lugares em que aparece é o livro *Manual de aritmética do mestre Sun*, escrito entre 287 d.C. e 473 d.C. O mesmo resultado é mencionado na *Aritmética* de Nicômaco de Gerasa. O algoritmo apenas codifica o método utilizado no problema da seção 2. Vamos analisá-lo com mais cuidado nesta seção.

Considere o sistema

$$\begin{aligned} x &\equiv a \pmod{m} \\ x &\equiv b \pmod{n}. \end{aligned} \tag{3.1}$$

Como na seção 2, vamos reescrever a primeira equação na forma $x = a + my$, onde y é um inteiro qualquer. Com isto podemos substituir x na segunda equação, obtendo $a + my \equiv b \pmod{n}$, ou ainda

$$my \equiv (b - a) \pmod{n}. \tag{3.2}$$

Já sabemos como resolver esta equação. Sabemos também que pode não ter solução. Para que (3.2) tenha solução é preciso que o máximo divisor comum entre m e n divida $b - a$.

Para garantir que isto vai acontecer basta assumir que $\text{mdc}(n, m) = 1$. Com isto $\overline{m}$ tem inverso em $\mathbb{Z}_n$. Chamando de $\overline{\alpha} \in \mathbb{Z}_n$ o inverso, a solução de (3.2) é $y \equiv \alpha(b - a) \pmod{n}$. Isto é $y = \alpha(b - a) + nz$, onde z é um número inteiro. Substituindo na equação que nos dá x em função de y, ficamos com

$$x = a + m\alpha(b - a) + mnz.$$

Mas $\overline{\alpha m} = \overline{1}$ em $\mathbb{Z}_n$. Logo, existe algum inteiro β tal que $1 - \alpha m = \beta n$. Assim

$$x = a(1 - m\alpha) + m\alpha b + mnz = a\beta n + m\alpha b + mnz.$$

Esta maneira de escrever a solução tem uma vantagem: os inteiros α e β são obtidos facilmente. De fato, $1 = \alpha m + \beta n$. Como estamos supondo que $\text{mdc}(m, n) = 1$, basta aplicar o algoritmo euclidiano estendido a m e n para obter α e β. Resumindo: se $\text{mdc}(m, n) = 1$, então o sistema (3.1) sempre tem como soluções os números $a\beta n + b\alpha m + kmn$, onde k é um inteiro qualquer.

Já vimos que uma equação linear pode ter mais de uma solução se o módulo for composto. Quantas soluções tem o sistema acima? Infinitas, se pensarmos em soluções inteiras, já que há uma solução para cada escolha de k. Mas vamos mais adiante. Digamos que x e y são dois inteiros que são solução do sistema (3.1). Então temos que $x \equiv a \pmod{m}$ e $y \equiv a \pmod{m}$. Como se tratam de duas equações *com mesmo módulo* podemos subtrair uma da outra. Obtemos $x - y \equiv 0 \pmod{m}$. Isto é, m divide $x - y$. Fazendo o mesmo com a segunda equação, concluímos que n divide $x - y$. Supondo que $\text{mdc}(m, n) = 1$, temos que mn divide $x - y$ pelo lema da seção 4 do Capítulo 2. Logo se x e y são soluções de (3.1) então $x \equiv y \pmod{mn}$. Em outras palavras, o sistema tem infinitas soluções inteiras, mas apenas uma solução em $\mathbb{Z}_{mn}$ quando $\text{mdc}(m, n) = 1$. Podemos reunir tudo isto em um teorema.

Teorema Chinês do Resto. *Sejam m e n inteiros positivos, primos entre si. O sistema*

$$x \equiv a \pmod{m}$$
$$x \equiv b \pmod{n}$$

sempre tem uma única solução em $\mathbb{Z}_{mn}$.

Para ver se de fato entendemos o conteúdo do teorema, é conveniente reinterpretá-lo na forma de uma tabela. Imagine que construímos uma tabela com mn casas. No alto da tabela, ao longo da horizontal, escrevemos os elementos de $\mathbb{Z}_m$ e à esquerda, ao longo da vertical, escrevemos os elementos de $\mathbb{Z}_n$. A casa da tabela que fica no encontro da coluna indexada por $\overline{a} \in \mathbb{Z}_m$, com a linha indexada por $\overline{b} \in \mathbb{Z}_n$ será ocupada pelo inteiro x tal que:
(1) $0 \leqslant x \leqslant mn - 1$
(2) $x \equiv a \pmod{m}$ e $x \equiv b \pmod{n}$.
Diremos, neste caso, que x tem *coordenadas* $(\overline{a}, \overline{b})$ na tabela.

O que é que o teorema chinês do resto nos diz sobre o preenchimento desta tabela? Supondo que $\text{mdc}(m, n) = 1$, o teorema afirma que toda casa da tabela é preenchida por algum inteiro no intervalo que vai de 0 a $mn - 1$, porque todos os sistemas do tipo (3.1) têm solução em $\mathbb{Z}_{mn}$. Além disso, como as soluções são únicas, duas casas com coordenadas distintas são preenchidas por elementos distintos de $\mathbb{Z}_{mn}$. Mas lembre-se que estamos supondo que os módulos são primos entre si.

Vamos desenhar a tabela quando $m = 4$ e $n = 5$. Observe que as classes da primeira linha e primeira coluna correspondem às coordenadas de cada uma das entradas da tabela.

	0	1	2	3
0	0	5	10	15
1	16	1	6	11
2	12	17	2	7
3	8	13	18	3
4	4	9	14	19

Observe que a tabela corresponde ao produto cartesiano $\mathbb{Z}_4 \times \mathbb{Z}_5$. Para preencher esta tabela não é necessário resolver 20 sistemas de congruências. Lembre-se que temos uma interpretação geométrica de $\mathbb{Z}_4$: as classes estão dispostas ao longo de uma circunferência. O mesmo vale para $\mathbb{Z}_5$. Assim a

tabela acima é como um mapa. Cortamos as circunferências correspondentes a $\mathbb{Z}_4$ e $\mathbb{Z}_5$ e estendemos o resultado sobre o plano.

Voltemos à tabela. É fácil achar as casas correspondentes a 0, 1, 2 e 3. Elas aparecem ao longo da 'diagonal' da tabela, que são as casas com coordenadas iguais. Mas onde colocar o 4? Não podemos continuar preenchendo a diagonal porque a tabela acabou; ou não? Claro que não. Pelo que vimos acima, os lados direito e esquerdo da tabela deveriam estar colados. Assim podemos continuar 'preenchendo ao longo da diagonal'. Só que o que deveria vir à direita, pula para a casa correspondente à esquerda. De modo semelhante, a tabela não acaba na linha de baixo. O lado de cima e o de baixo coincidem também. Portanto, da última linha pulamos para a casa uma coluna à direita, na primeira linha.

Para você entender melhor o que estamos fazendo, desenhamos a seguir a tabela preenchida apenas com os números de 0 a 5. Para fazer isto tivemos que efetuar dois 'pulos'. O primeiro da margem direita para a esquerda, nos permite escrever o 4 na última linha. O segundo da margem de baixo para a de cima, nos permite escrever o 5 na segunda coluna. Continuando assim até usar todos os números disponíveis, obtemos a primeira tabela acima.

	0	1	2	3
0	0	5		
1		1		
2			2	
3				3
4	4			

Uma pergunta: qual a verdadeira forma da tabela, antes de termos cortado e projetado no plano? Colando o lado inferior e o superior, temos um cilindro. Precisamos agora colar as circunferências que formam as extremidades do cilindro. Fazendo isto, obtemos uma superfície parecida com uma rosquinha de padaria ou uma câmara de ar cheia. Esta superfície chama-se um *toro*. Assim a verdadeira tabela $\mathbb{Z}_m \times \mathbb{Z}_n$ só pode ser representada sobre a superfície de um toro.

4. Módulos não co-primos

Até agora tratamos em detalhes apenas o caso em que os módulos são primos entre si. Na verdade este é o único caso que usaremos adiante, por isso não faremos uma análise detalhada do caso geral. Isto porém não nos impede

de resolver sistemas em que os módulos não são primos entre si. Para um tal sistema, teremos que analisar o que acontece às equações à medida que formos aplicando o algoritmo. Um exemplo será suficiente para esclarecer o que ocorre no caso geral. Considere o sistema:

$$x \equiv 3 \pmod{12}$$
$$x \equiv 19 \pmod{8}.$$

A primeira equação dá lugar a $x = 3 + 12y$. Substituindo isto na segunda equação, temos $12y \equiv 16 \pmod 8$. Como $\text{mdc}(12, 8) = 4$ divide 16, esta última equação tem solução. Na verdade, ela é equivalente a $12y - 8z = 16$. Cancelando 4 de ambos os membros, obtemos $3y - 2z = 4$. Ou ainda $3y \equiv 4 \pmod 2$. Mas $3 \equiv 1 \pmod 2$, donde $y \equiv 0 \pmod 2$. Logo $y = 2k$, para algum inteiro k. Substituindo isto na expressão de x em função de y, obtemos $x = 3+24k$. Observe que neste caso temos uma única solução, só que módulo 24 e não módulo $8 \cdot 12 = 96$. Qual a relação entre 24 e os módulos 8 e 12? A resposta pode ser encontrada no exercício 5.

Em geral, quando o máximo divisor comum dos módulos é diferente de 1, nem todos os sistemas que podemos escrever têm solução. Se pensarmos em termos da representação gráfica, isto significa que nem todas as casas da tabela serão preenchidas.

Mais uma vez não é necessário resolver nenhum sistema para preencher a tabela. Basta ir preenchendo a diagonal, e lembrando que a tabela habita a superfície de um toro. Fazendo isto, quando os módulos m e n *não* são primos entre si, voltamos à casa de coordenadas $(\overline{0}, \overline{0})$ antes de esgotar os números de 0 a $mn - 1$. É por isso que há casas que não são preenchidas.

A tabela no caso em que $m = 4$ e $n = 6$ é a seguinte:

	0	1	2	3
0	0		6	
1		1		7
2	8		2	
3		9		3
4	4		10	
5		5		11

5. Potências, novamente

Existe uma versão do teorema chinês do resto para mais de duas equações, vamos enunciá-la e deixar a você a tarefa de escrever a demonstração, que segue de perto o padrão do caso de duas equações.

Teorema Chinês do Resto. *Sejam* $n_1, \ldots, n_k$ *inteiros positivos dois a dois primos entre si. Então o sistema*

$$\begin{aligned} x &\equiv a_1 \pmod{n_1} \\ x &\equiv a_2 \pmod{n_2} \\ &\vdots \\ x &\equiv a_k \pmod{n_k} \end{aligned}$$

tem uma única solução em $\mathbb{Z}_{n_1 \ldots n_k}$.

Dizer que os inteiros $n_1, \ldots, n_k$ são dois a dois primos entre si significa que $\mathrm{mdc}(n_i, n_j) = 1$ sempre que os índices i e j são distintos. No caso de três módulos n_1, n_2, n_3 teremos $\mathrm{mdc}(n_1, n_2) = \mathrm{mdc}(n_1, n_3) = \mathrm{mdc}(n_2, n_3) = 1$.

Podemos aplicar o *Teorema Chinês do Resto* para simplificar o cálculo de potências módulo n em alguns casos especiais. Suponhamos que $n = p_1 \ldots p_k$ onde $p_1 < \cdots < p_k$ são números primos. Assim, estamos supondo que, na fatoração de n, cada fator primo aparece com multiplicidade 1. Neste caso fica muito fácil calcular a forma reduzida de a^m módulo n. Em primeiro lugar, usamos o teorema de Fermat para achar a forma reduzida de a^m módulo cada um dos primos $p_1, \ldots, p_k$ separadamente. Digamos que

$$\begin{aligned} a^m &\equiv r_1 \pmod{p_1} \quad \text{e} \quad 0 \leqslant r_1 \leqslant p_1 \\ a^m &\equiv r_2 \pmod{p_2} \quad \text{e} \quad 0 \leqslant r_2 \leqslant p_2 \\ &\cdots \\ a^m &\equiv r_k \pmod{p_k} \quad \text{e} \quad 0 \leqslant r_k \leqslant p_k \end{aligned}$$

Para achar a forma reduzida de a^m módulo n basta resolver o sistema

$$\begin{aligned} x &\equiv r_1 \pmod{p_1} \\ x &\equiv r_2 \pmod{p_2} \\ &\vdots \\ x &\equiv r_k \pmod{p_k}. \end{aligned}$$

Observe que este sistema sempre tem solução, já que os módulos são primos distintos e, portanto, o máximo divisor comum entre dois quaisquer dentre eles é sempre 1. Além disso, o teorema nos garante que o sistema tem uma única solução r módulo $p_1 \dots p_k = n$. Logo $a^m \equiv r \pmod{n}$, e obtivemos o que queríamos.

Vejamos um exemplo numérico. Digamos que queremos calcular a forma reduzida de 2^{6754} módulo 1155. Fatorando 1155 vemos que é igual a $3 \cdot 5 \cdot 7 \cdot 11$; todos primos com multiplicidade 1. Aplicando o Teorema de Fermat a cada um destes primos, temos que

$$\begin{aligned} 2^{6754} &\equiv 1 \pmod{3} \\ 2^{6754} &\equiv 4 \pmod{5} \\ 2^{6754} &\equiv 2 \pmod{7} \\ 2^{6754} &\equiv 5 \pmod{11}. \end{aligned}$$

Precisamos, portanto resolver o sistema

$$\begin{aligned} x &\equiv 1 \pmod{3} \\ x &\equiv 4 \pmod{5} \\ x &\equiv 2 \pmod{7} \\ x &\equiv 5 \pmod{11}. \end{aligned}$$

Usando o algoritmo chinês temos que $x = 1 + 3y$ que, substituído na segunda equação, dá

$$1 + 3y \equiv 4 \pmod{5} \quad \text{isto é} \quad y \equiv 1 \pmod{5},$$

já que 3 é inversível módulo 5 e pode ser cancelado dos dois membros da equação. Assim $x = 4 + 15z$. Substituindo isto na terceira equação e resolvendo-a obtemos $z \equiv 5 \pmod{7}$; ou seja $x = 79 + 105t$. Finalmente substituindo isto na última equação, teremos $t \equiv 6 \pmod{11}$, o que dá $x = 709 + 1155u$. Concluímos que $2^{6754} \equiv 709 \pmod{1155}$. Para realmente apreciar as vantagens deste método, experimente refazer os cálculos sem usá-lo.

6. Partilha de senhas

Benjamin Franklin, o inventor do pára-raios, disse uma vez que *'três pessoas só conseguem guardar um segredo, se duas delas já estiverem mortas'*. Nesta seção apresentaremos um método que permite compartilhar um se-

gredo entre várias pessoas–que ainda estão vivas. Mais precisamente, imagine que o cofre de um banco é aberto através de uma senha. Um certo número de funcionários do banco tem acesso ao cofre. Mas o banco deseja, por segurança, que não seja possível abrir o cofre quando há menos de três destes funcionários presentes na agência. Digamos que há na agência vinte funcionários com acesso ao cofre. Como garantir que não será possível abrir o cofre a menos que haja, pelo menos, três destes funcionários presentes na agência?

Vamos generalizar o problema acima e delimitá-lo com mais cuidado. Para abrir o cofre do banco é necessário conhecer a senha, que é um número s. Queremos partilhar s entre n pessoas. A cada pessoa vai ser dado um elemento (sua 'parte' da senha) de um conjunto $\mathbb{S}$ de n *pares* de inteiros positivos, de modo que, para um inteiro positivo $k \leqslant n$, previamente escolhido temos:

(1) qualquer subconjunto de $\mathbb{S}$ com k elementos permite determinar s facilmente;

(2) é muito difícil determinar s conhecendo menos de k elementos de $\mathbb{S}$.

A inspiração para a construção do conjunto $\mathbb{S}$ vem do teorema chinês do resto. Começamos escolhendo um conjunto $\mathcal{L}$ de n inteiros positivos, dois a dois primos entre si. Seja N o produto dos k *menores* números de $\mathcal{L}$ e M o produto dos $k-1$ *maiores* números de $\mathcal{L}$. Diremos que este conjunto tem *limiar* k se

$$N > s > M.$$

Observe que esta condição implica que o produto de k ou mais elementos de $\mathcal{L}$ é sempre *maior* que N e o produto de menos de k elementos é sempre *menor* que M. O conjunto $\mathbb{S}$ será formado pelos pares da forma (m, s_m) onde $m \in \mathcal{L}$ e s_m é a forma reduzida de s módulo m. Note que o fato de termos um conjunto com limiar $k \geqslant 1$ implica que $s > m$, para qualquer $m \in \mathcal{L}$. Portanto sempre temos $s_m < s$, qualquer que seja $m \in \mathcal{L}$.

Suponhamos que mais de k funcionários se encontram no banco. Isto equivale a dizer que são conhecidos t dentre os pares de $\mathbb{S}$, onde $t \geqslant k$. Denotaremos estes pares por $(m_1, s_1), \ldots, (m_t, s_t)$. Vamos resolver o sistema de congruências

$$\begin{aligned} x &\equiv s_1 \pmod{m_1} \\ x &\equiv s_2 \pmod{m_2} \\ &\vdots \\ x &\equiv s_t \pmod{m_t} \end{aligned} \tag{6.1}$$

obtendo x_0 como solução. De acordo com o teorema chinês do resto,

$$x_0 \equiv s \pmod{m_1 \dots m_t}.$$

Mas será que são iguais? É aqui que entra o fato da seqüência de módulos ter limiar k. Sabemos que, como $t \geqslant k$,

$$m_1 \dots m_t \geqslant N > s.$$

Pelo teorema chinês do resto, o sistema (6.1) tem uma *única* solução menor que $m_1 \dots m_t$. Mas s também é solução do sistema e $s < m_1 \dots m_t$. Logo $s = x_0$.

Nada nos impede de resolver um sistema semelhante para o caso em que $t < k$. O problema é que o produto de menos de k módulos de $\mathcal{L}$ é sempre menor que s. Assim a solução do sistema é um número congruente a s, mas não pode ser igual a s. É claro que sempre é possível achar s fazendo uma busca. De fato, sabemos que $M < s < N$ e que s satisfaz o sistema (6.1), só que agora com $t < k$. Digamos que achamos uma solução x_0 do sistema. Como $x_0 < M < s$, não achamos s. Mas o sistema (6.1) também é satisfeito por s, logo

$$s = x_0 + y \cdot (m_1 \dots m_t)$$

onde y é um inteiro positivo. Como

$$N > s > M > x_0,$$

temos que

$$\frac{M - x_0}{m_1 \dots m_t} \leqslant y = \frac{s - x_0}{m_1 \dots m_t} \leqslant \frac{N - x_0}{m_1 \dots m_t}.$$

Isto significa que precisamos fazer uma busca para achar o valor correto de y entre, pelo menos,

$$d = \left[\frac{N - M}{m_1 \dots m_t} \right]$$

inteiros. Escolhendo os módulos de modo que d seja muito grande, fica praticamente impossível encontrar s através de uma busca.

Resta um problema: é sempre possível escolher um conjunto $\mathcal{L}$ satisfazendo a todas estas condições? A resposta é sim, mas a solução depende de resultados sobre a distribuição de primos que estão fora do nosso alcance nestas notas. Os detalhes podem ser encontrados em [**Kranakis 1986, Cap. 1, seção 5**].

Na prática os dados iniciais do problema são o número total de funcionários da agência e o número mínimo de funcionários que têm que estar

presentes para que o cofre possa ser aberto. O primeiro determina quantos elementos o conjunto $\mathcal{L}$ tem que ter. O segundo determina qual é o limiar k de $\mathcal{L}$. Com estes dados, escolhemos um conjunto $\mathcal{L}$ de limiar k–esta é a parte do problema que não discutimos em detalhe. Com isto podemos calcular M e N como acima. Só agora s será escolhido (de maneira aleatória) no intervalo entre M e N. Chegados a este ponto, já temos todos os dados necessários para calcular $\mathbb{S}$, que nos diz quem são as senhas a serem distribuídas. É claro que a segurança do sistema se baseia no fato de que, quanto maior o valor de k, mais improvável será que haja k funcionários desonestos no banco. Se todos os funcionários forem desonestos, estamos perdidos: nenhum sistema é totalmente à prova de desonestidade.

Vejamos um exemplo numérico bem simples. Digamos que há apenas 5 funcionários e que pelo menos 2 têm que estar presentes para que o cofre possa ser aberto. Logo o conjunto $\mathcal{L}$ deve ter 5 elementos e seu limiar deve ser 2. Uma escolha possível, usando apenas primos pequenos, é

$$\mathcal{L} = \{11, 13, 17, 19, 23\}.$$

De fato, o produto dos 2 menores inteiros no conjunto é $N = 11 \cdot 13 = 143$. Por outro lado, M é o produto dos $k-1$ maiores elementos de $\mathcal{L}$. Como $k = 2$, temos que M é *igual* ao maior elemento de $\mathcal{L}$; logo $M = 23$. Portanto $\mathcal{L}$ tem realmente limiar 2. O valor de s pode ser escolhido como sendo qualquer inteiro no intervalo que vai de 23 a 143. Digamos que $s = 30$. Então

$$\mathbb{S} = \{(11, 19), (13, 17), (17, 13), (19, 11), (23, 7)\}.$$

Finalmente, o que acontece se os funcionários que têm senhas $(17, 13)$ e $(23, 7)$ estão no banco? Para obter a senha s do cofre é preciso resolver o sistema

$$\begin{aligned} x &\equiv 13 \pmod{17} \\ x &\equiv 7 \pmod{23}. \end{aligned}$$

A solução geral do sistema é $x = 30 + 391k$, onde k é um inteiro positivo. Isto é $x \equiv 30 \pmod{391}$. Assim determinamos que $s = 30$; que é o valor correto.

7. Exercícios

1. Um problema de Yih-hing (717 d.C.): ache a solução do sistema: $x \equiv 1 \pmod 2$, $x \equiv 2 \pmod 5$ e $x \equiv 5 \pmod{12}$.

2. Um outro velho problema chinês:

Três fazendeiros cultivavam juntos todo o seu arroz e o dividiam igualmente entre si no tempo da colheita. Um certo ano cada um deles foi a um mercado diferente vender o seu arroz. Cada um destes mercados só comprava arroz em múltiplos de um peso padrão, que diferia em cada um dos mercados. O primeiro fazendeiro vendeu o seu arroz em um mercado onde o peso padrão era 87 Kg. Ele vendeu tudo o que podia e voltou para casa com 18 Kg de arroz. O segundo fazendeiro vendeu todo o arroz que podia em um mercado cujo peso padrão era de 170 Kg e voltou para casa com 58 Kg. O terceiro fazendeiro vendeu todo o arroz que podia em um mercado cujo peso padrão era de 143 Kg e voltou (ao mesmo tempo que os outros dois) com 40 Kg. Qual a quantidade mínima de arroz que eles podem ter cultivado, no total?

3. Determine o menor inteiro positivo que deixa resto 2 na divisão por 5, resto 4 na divisão por 7 e resto 5 na divisão por 11.

4. Na encenação de uma batalha, duas tropas se enfrentam, posicionando-se, atirando com festim, e recarregando seus mosquetes, cada uma a sua vez. Cada lado começa com o mesmo número de cartuchos. Uma tropa tem 100 mosquetes e, depois de atirar tantos tiros de festim quanto possíveis, lhe sobram 13 cartuchos. A outra tropa tem 67 mosquetes, e ao fim da exibição, sobram-lhe 32 cartuchos. Supondo que a cada salva de tiros todos os soldados de cada lado atiraram exatamente uma vez, determine o número mínimo de cartuchos com que cada tropa iniciou a exibição.

5. Mostre que o sistema

$$x \equiv a \pmod{m}$$
$$x \equiv b \pmod{n}$$

não pode ter mais que uma solução módulo o mínimo múltiplo comum entre m e n. Observe que não estamos supondo que m e n são primos entre si.

6. Determine o resto da divisão de 2^{45632} e de 3^{54632} por 12155.

7. Resolva a equação $x^2 + 42x + 21 \equiv 0 \pmod{105}$.

Sugestão: Fatore 105, resolva a equação módulo cada fator primo e 'cole' as soluções usando o teorema chinês do resto.

8. Qual a mais longa seqüência de *primos consecutivos* com limiar 3 e primeiro elemento 11? E com limiar 4?

9. Sejam p e q primos distintos e $n = pq$. Digamos que conhecemos soluções para as equações $x^2 \equiv a \pmod{p}$ e $x^2 \equiv a \pmod{q}$. Mostre como usar o teorema chinês do resto para achar uma solução de $x^2 \equiv a \pmod{n}$. Compare com o método da seção 5 e com o exercício 7.

10. Sejam p e q primos distintos e $n = pq$. Suponhamos que ambos os primos deixam resto 3 na divisão por 4. Escreva um programa que, tendo como entrada p, q e a, calcula uma solução de $x^2 \equiv a \pmod{n}$. A restrição imposta aos primos facilita a solução das equações $x^2 \equiv a \pmod{p}$ e $x^2 \equiv a \pmod{q}$; veja exercício 17 do capítulo 5. Este exercício é o terceiro de uma seqüência que termina com o exercício 8 do capítulo 11.

8

Grupos

Neste capítulo introduzimos grupos e subgrupos e provamos o teorema de Lagrange. A noção de grupo faz parte da taxonomia dos objetos matemáticos. Em outras palavras, sua função é organizar objetos matemáticos que têm características comuns. É necessário, portanto, conhecer um grande número de exemplos para realmente entender o que se esconde por trás da definição de grupo.

1. Definição e exemplos

Um *grupo* é constituído de dois ingredientes básicos: um *conjunto* e uma *operação* definida neste conjunto. Vamos chamar o conjunto de G e a operação de $*$. Por operação estamos entendendo uma regra que a cada dois elementos $a, b \in G$ associa um terceiro elemento $a * b$ que também está em G.

A associação 'conjunto e operação' é extremamente comum em matemática. Alguns dos exemplos familiares incluem: os números naturais e a soma, os inteiros e a soma, os inteiros e o produto, os racionais e a soma, os racionais e o produto, os vetores no espaço e a soma de vetores, os vetores no espaço e o produto vetorial e muitos, muitos mais.

Entretanto nem toda associação 'conjunto e operação' constitue um grupo. Para termos um grupo é necessário que a operação satisfaça algumas propriedades.

Um conjunto G onde está definida uma operação $*$ é um *grupo* se a operação satisfaz as seguintes propriedades:

(1) **Associatividade:** dados $a, b, c \in G$ temos que

$$a * (b * c) = (a * b) * c.$$

(2) **Elemento neutro:** existe um elemento $e \in G$ tal que para todo $a \in G$

temos

$$a * e = e * a = a.$$

(3) **Elemento inverso:** dado um elemento $a \in G$ qualquer, existe um elemento $a' \in G$ (o **inverso** de a) tal que

$$a * a' = a' * a = e.$$

Observe que *não* estamos exigindo que a operação de um grupo seja comutativa; isto é, que $a * b = b * a$ para quaisquer elementos $a, b \in G$. A razão para isto é puramente pragmática: há muitos grupos interessantes que não satisfazem esta propriedade. Um grupo cuja operação é comutativa é chamado de *abeliano*.

Há muitos exemplos importantes da associação 'conjunto e operação' que *não* satisfazem as propriedades acima e, portanto, não são grupos. Por exemplo, a soma de números naturais é associativa e tem elemento neutro (o zero). Entretanto o único natural a ter inverso para a soma é o zero; porque os números negativos não são naturais. Outro exemplo, ainda mais drástico: o conjunto dos vetores no espaço com o produto vetorial não satisfaz sequer a propriedade associativa.

Em contrapartida, a ubiqüidade dos grupos é inegável. Os inteiros, racionais, reais e complexos são grupos para a soma. Os racionais, reais e complexos *não nulos* são grupos para o produto. Para qualquer inteiro positivo n, o conjunto $\mathbb{Z}_n$ com a soma é um grupo. As matrizes formam um grupo para a soma de matrizes. As matrizes inversíveis (isto é, de determinante não nulo) formam um grupo para o produto de matrizes. O mesmo ocorre para as matrizes de determinante 1. Estes dois últimos são exemplos de grupos onde a operação *não* é comutativa.

O número de elementos de um grupo é a sua *ordem*. A maior parte dos grupos mencionados anteriormente tem ordem infinita. O único exemplo que vimos de grupo finito foi $\mathbb{Z}_n$ com a soma; este grupo tem ordem n. Um outro grupo finito fácil de descrever é $\{-1, 1\}$ com a operação de produto de inteiros; neste caso a ordem é 2. Veremos mais exemplos de grupos finitos nas seções seguintes.

2. Simetrias

Uma das aplicações mais interessantes dos grupos é ao estudo de simetrias. Pode-se até dizer que os grupos são a tradução matemática da idéia de simetria. Os grupos de simetria são fundamentais em geometria, em cristalografia e em física. Curiosamente, a noção de *simetria* é relativamente difícil

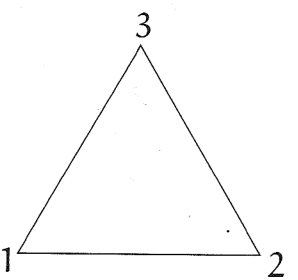
de descrever com precisão. Uma simetria de uma figura geométrica, por exemplo, pode ser descrita como uma 'transformação que, quando aplicada à figura, não altera o seu aspecto'. Ou melhor ainda: imagine que você fecha os olhos enquanto aplico alguma transformação à figura. Se, abrindo os olhos, você não for capaz de dizer se fiz alguma coisa à figura ou não, então esta transformação é uma simetria.

Vamos tentar descrever as simetrias de um triângulo eqüilátero. Em primeiro lugar temos três rotações possíveis no sentido anti-horário: de 120°, 240° e 360°. Esta última coincide com uma rotação de 0°. Além disso há três reflexões. Cada uma destas tem por 'eixo' (ou 'espelho') uma das mediatrizes do triângulo. Isto nos dá um total de seis elementos. Pode-se provar que estas são todas as simetrias do triângulo eqüilátero.

Já temos o conjunto, falta a operação. A operação neste caso é a composição de transformações; isto é, realizamos uma transformação em seguida à outra. Como a composição de funções é sempre associativa, a primeira propriedade da operação de um grupo é imediatamente satisfeita neste caso. O elemento neutro é a rotação de 0°, que não faz nada ao triângulo.

E quanto aos inversos? O inverso da rotação de 120° é a rotação de 240°; e vice-versa. Cada reflexão é inversa dela própria. Logo todas as transformações descritas acima têm inversa; e temos um grupo. Este grupo é conhecido como D_3. Observe que o grupo D_3 tem ordem 6.

Vamos numerar os vértices do triângulo eqüilátero: 1 e 2 serão os vértices da base e 3 o outro vértice.



Podemos então descrever as transformações anteriores como permutações dos vértices. Por exemplo, a rotação de 120° leva cada vértice no adjacente na direção anti-horária. Existe uma notação prática para descrever esta permutação:

$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}.$$

Esta é a transformação que, quando aplicada ao triângulo, leva o vértice 1 no

2, o 2 no 3 e o 3 no 1: na linha de cima da matriz sempre escrevemos os vértices na ordem 1, 2, 3; na linha de baixo escrevemos a imagem correspondente de cada vértice. Um outro exemplo. A reflexão em torno da mediatriz do vértice 3 é dada por:

$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}.$$

Se ρ é a rotação de $120°$, então

$$\rho^2 = \rho \cdot \rho$$

é a rotação de $240°$ e $\rho^3 = e$ é o elemento neutro (rotação de $360°$). Digamos que σ seja uma das reflexões, então $\sigma^2 = e$. Desejamos agora identificar o elemento $\sigma\rho$. Observe que $\sigma\rho$ não pode ser igual a ρ^2. De fato, multiplicando a equação

$$\sigma\rho = \rho^2$$

à direita por ρ^2 e usando que $\rho^3 = e$, teríamos que $\sigma = \rho$, o que é falso. Podemos mostrar, de maneira semelhante, que $\sigma\rho \neq e$ e que $\sigma\rho \neq \rho$. Logo $\sigma\rho$ não pode ser uma rotação. Então $\sigma\rho$ é uma reflexão. Mas $\sigma\rho \neq \sigma$, já que $\sigma\rho = \sigma$ nos dá que $\rho = e$, o que não é verdade. Portanto $\sigma\rho$ é uma reflexão diferente de σ.

Vamos denotar por σ_3 a reflexão que fixa o vértice 3, descrita acima. Então ρ move o vértice 1 para a posição do vértice 2, ao passo que σ_3 move o vértice 2 para a posição de 1. Portanto $\sigma_3\rho(1) = 1$; isto é, o vértice 1 *não* se move sob a ação de $\sigma_3\rho$. Assim descobrimos que $\sigma_3\rho$ é a reflexão que deixa o vértice 1 parado. Vamos denotá-la por σ_1. Podemos calcular $\sigma_3\rho$ usando a notação sugerida acima. Temos que

$$\sigma_3 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix} \quad \text{e} \quad \rho = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}.$$

Antes de executar a conta, observe que $\sigma_3\rho(1)$ significa que ρ é aplicado a 1 *antes* de σ_3, assim

$$1 \xrightarrow{\rho} 2 \xrightarrow{\sigma_3} 1.$$

Portanto

$$\sigma_3\rho = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}\begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix} = \sigma_1.$$

Usando apenas as propriedades básicas de grupos podemos determinar várias outras relações entre os elementos de D_3 a partir de $\sigma_3\rho = \sigma_1$. Por

exemplo, se G é um grupo com uma operação $*$ e $x, y \in G$, então o inverso de $x * y$ é $y' * x'$. Para verificar isto, basta multiplicar os dois elementos

$$(x * y) * (y' * x') = x * (y * y') * x' = x * e * x' = x * x' = e.$$

Usando isto, concluímos que

$$(\sigma_3\rho)' = \rho^2\sigma_3.$$

Mas já vimos que $\sigma_3\rho = \sigma_1$. Como $\sigma_1^2 = e$, temos também que $\rho^2\sigma_3 = \sigma_1$. Portanto,

$$\sigma_3\rho = \sigma_1 = \rho^2\sigma_3 \neq \rho\sigma_3.$$

Em particular, o grupo não é abeliano.

Várias outras relações seguem de $\sigma_3\rho = \sigma_1$. Multiplicando esta relação à esquerda por σ_3 e usando que

$$\sigma_3^2 = e, \quad \text{temos} \quad \rho = \sigma_3\sigma_1.$$

Já multiplicando *à direita* por ρ^2 obtemos $\sigma_3 = \sigma_1\rho^2$. Observe que como a operação do grupo não é comutativa, é necessário especificar se estamos multiplicando ambos os membros da equação à direita ou à esquerda por um determinado elemento.

Continuando desta maneira, podemos encontrar a *tabela de multiplicação* deste grupo. Entendemos por este nome a tabela construída escrevendo o produto $x * y$ na intersecção da linha que começa com o elemento x com a coluna que começa com o elemento y. Para o grupo D_3 a tabela é a seguinte.

	e	ρ	ρ^2	σ_1	σ_2	σ_3
e	e	ρ	ρ^2	σ_1	σ_2	σ_3
ρ	ρ	ρ^2	e	σ_3	σ_1	σ_2
ρ^2	ρ^2	e	ρ	σ_2	σ_3	σ_1
σ_1	σ_1	σ_2	σ_3	e	ρ	ρ^2
σ_2	σ_2	σ_3	σ_1	ρ^2	e	ρ
σ_3	σ_3	σ_1	σ_2	ρ	ρ^2	e

Observe que não há elementos repetidos em uma determinada linha ou coluna desta tabela. Isto é um fato geral, que vale para a tabela de qualquer grupo. Para provar este fato imagine que temos um grupo finito G munido de uma operação $*$. Se $a \in G$, a linha correspondente a este elemento é preenchida com os valores dos produtos $a * x$, para cada $x \in G$. Para que

houvesse dois elementos iguais nesta linha seria necessário que $a * x = a * y$ com $x \neq y$. Porém de $a * x = a * y$ deduzimos que

$$x = a' * (a * x) = a' * (a * y) = y.$$

Logo todos os elementos de uma determinada linha da tabela de G são distintos. A demonstração para as colunas é análoga.

De uma maneira geral, o grupo de simetrias de um polígono *regular* de n lados, denotado por D_n, tem ordem $2n$ e é gerado pela rotação ρ de $360/n$ graus e por qualquer uma das reflexões. Chamando de σ uma das reflexões, temos que a relação

$$\sigma\rho = \rho^{n-1}\sigma$$

é sempre verificada. O grupo D_n é conhecido como *grupo diedral* de ordem $2n$. Para provar que esta descrição corresponde de fato ao grupo de simetrias de um polígono regular é preciso recorrer à álgebra linear. Para mais detalhes consulte o livro [**Artin 1991, Cap. 5**].

3. Interlúdio

A teoria dos grupos é uma aquisição relativamente recente da matemática, que surgiu no âmbito da teoria de equações. Muito antes de Cristo, os babilônios já conheciam a fórmula para resolver a equação do segundo grau. Este tipo de problema algébrico não interessou muito aos gregos. Já os árabes o estudaram com cuidado, tentando encontrar maneiras de resolver equações do terceiro grau. Mas foi na Itália Renascentista que as fórmulas para resolver equações polinomiais de terceiro e quarto graus foram descobertas.

Na história desta descoberta não faltam as intrigas típicas de um drama hollywoodiano. Tudo começou com Scipione del Ferro, professor de matemática em Bolonha, uma das mais antigas universidades medievais. Não se sabe exatamente quando del Ferro descobriu como resolver a equação cúbica, mas antes de morrer (por volta de 1526), revelou o método a seu estudante Antonio Maria Fior. Nesta época, competições eram comuns entre os letrados e Fior resolveu desafiar um professor de matemática de Veneza, Niccolò Tartaglia ('o gago') para uma disputa. Cada competidor propôs 30 problemas a serem resolvidos em um tempo fixado; o perdedor teria que pagar 30 banquetes. Tartaglia redescobriu a fórmula da cúbica e resolveu todos os problemas que lhe haviam sido propostos; Fior não resolveu nenhum!

O triunfo de Tartaglia fez com que fosse convidado a visitar Gerolamo Cardano, que era um misto de médico, matemático, filósofo e astrólogo. Tartaglia contou a Cardano os detalhes da sua fórmula, mas o fez jurar que não

a divulgaria. Cardano estendeu o método de Tartaglia e finalmente o publicou, em flagrante perjúrio, em sua *Ars Magna* de 1545. Além da solução da cúbica (com a atribuição a Tartaglia), o livro trazia um método para reduzir a equação completa de quarto grau a uma cúbica. Este último resultado Cardano atribui a Lodovico Ferrari, seu amigo e secretário.

Durante os 400 anos seguintes muitos matemáticos tentaram obter, sem sucesso, métodos semelhantes para resolver equações de grau maior que 4. O que se procurava era uma fórmula em que as raízes fossem escritas a partir dos coeficientes da equação em termos das quatro operações fundamentais além, é claro, da extração de raízes. Finalmente em 1824 um jovem matemático norueguês, Niels Abel, deu uma demonstração correta de que *não existe* uma tal fórmula para equações de grau maior que 4. Abel contribuiu também para muitas outras áreas da matemática, como análise e geometria algébrica. Seu nome está associado aos grupos abelianos, às funções abelianas e ao teorema de Abel sobre convergência de séries. Tudo isto Abel realizou até 1829; quando morreu de tuberculose aos 27 anos. É tal a fama de Abel, que uma estátua em sua homenagem foi erigida em 1908 no Parque Real, no centro de Oslo.

O ponto final neste problema da teoria de equações foi posto pelo matemático francês Evariste Galois. Dada uma equação polinomial, Galois mostrou que é possível associar a esta equação um certo grupo finito. Das propriedades deste grupo é possível deduzir se a equação pode ou não ser resolvida usando as quatro operações e a extração de raízes. Isto dá um critério definitivo para determinar se uma equação pode ou não ser resolvida algebricamente.

A vida de Galois foi ainda mais trágica que a de Abel. Seu pai suicidou-se por razões políticas; seus trabalhos foram considerados incompreensíveis pelos matemáticos da Academia de Ciências e sua admissão à Escola Politécnica foi rejeitada. Diz-se que, enfurecido Galois teria jogado o apagador na cabeça de um dos examinadores do 'vestibular' da Politécnica. Republicano em uma França monarquista, Galois foi expulso da Escola Normal por questões políticas. Em outra ocasião propôs um brinde ao rei, durante um banquete, enquanto brandia um punhal. Foi preso no dia seguinte. Durante uma epidemia de cólera em 1832, teve que ser transferido da prisão para um hospital. Lá apaixonou-se por uma garota e, aparentemente por causa disto, foi desafiado a um duelo. Na noite anterior ao duelo, escreveu uma carta de despedida a seu amigo Auguste Chevalier. Nesta carta, após resenhar suas descobertas, ele conclui:

Tu pedirás publicamente a Gauss ou Jacobi para que dêem sua opinião, não sobre a verdade, mas sobre a importância destes teoremas. Mais tarde haverá, espero, pessoas que verão proveito em decifrar esta confusão.
Ferido mortalmente no duelo, foi abandonado no campo de honra. Só horas depois foi encontrado e levado ao hospital. A única pessoa de sua família avisada à tempo foi seu irmão mais novo. Sabendo que ia morrer, ainda achou forças para confortar o irmão dizendo *'não chore; preciso de toda a minha coragem para morrer aos vinte anos'*.

E. Galois (1811 – 1832).

Foi Joseph Liouville quem finalmente 'decifrou a confusão' e revelou a riqueza dos resultados descobertos por Galois. Em 1846 Liouville publicou toda a obra matemática deixada por Galois (inclusive a carta a Chevalier), um total de apenas 64 páginas impressas. A carta a Chevalier foi descrita por um grande matemático do século XX da seguinte maneira:
Esta carta, se julgada pela novidade e profundidade das idéias que contém, é talvez o texto mais substancial de toda a literatura da humanidade.

Não admira que Galois seja considerado um dos fundadores da álgebra moderna. Foi a partir de sua obra que a noção de grupo (o nome foi inventado por ele) ganhou a importância que tem hoje. Mais detalhes sobre a obra de Galois podem ser encontrados em [**Gonçalves 1979**] ou [**Artin 1991**]. Para uma história mais abrangente dos acontecimentos discutidos acima consulte [**Boyer 1994**] ou [**van der Waerden 1985**].

4. Grupos Aritméticos

Não podemos perder de vista que nosso interesse último é o estudo de propriedades aritméticas, sobretudo de caráter *multiplicativo*; isto é, propriedades dos inteiros que têm a ver com primos e fatoração. É para estudá-las melhor que introduzimos grupos. Mas qual o grupo que desejamos estudar?

Comecemos com um inteiro positivo n. Denotamos por $U(n)$ o conjunto dos elementos inversíveis de $\mathbb{Z}_n$; veja a seção 7 do capítulo 4. Em outras palavras,

$$U(n) = \{\overline{a} \in \mathbb{Z}_n : \mathrm{mdc}(a, n) = 1\}.$$

Este conjunto é um grupo para a operação de multiplicação de classes de $\mathbb{Z}_n$. Dada a importância do exemplo, vamos verificar esta afirmação com cuidado.

Em primeiro lugar, sabemos que o produto de dois elementos de $\mathbb{Z}_n$ é um elemento de $\mathbb{Z}_n$: mas nada nos garante que isto também vale para $U(n)$? Isto é, precisamos saber se, quando multiplicamos dois elementos de $U(n)$, o resultado é um elemento de $U(n)$–do contrário não temos uma operação em $U(n)$.

Em outras palavras, precisamos verificar que o produto de dois elementos inversíveis de $\mathbb{Z}_n$ é um elemento inversível de $\mathbb{Z}_n$. Já fizemos isto na seção 7 do capítulo 4, mas o argumento é simples e não custa repeti-lo. Digamos que $\overline{a}$ e $\overline{b}$ são elementos de $U(n)$ e que seus inversos são $\overline{a'}$ e $\overline{b'}$, respectivamente. Então $\overline{ab}$ é inversível e seu inverso é $\overline{a'b'}$, como é fácil verificar multiplicando estes dois elementos

$$\overline{ab} \cdot \overline{a'b'} = \overline{aa'} \cdot \overline{bb'} = \overline{1}.$$

Portanto temos um conjunto $U(n)$, onde está definida uma operação, o produto de classes. Precisamos verificar que esta operação satisfaz as propriedades requeridas. A associatividade é fácil, porque já sabemos que o produto em $\mathbb{Z}_n$ é associativo. O elemento neutro é $\overline{1}$, que é inversível em $\mathbb{Z}_n$, e portanto está em $U(n)$. Que cada elemento de $U(n)$ tem inverso é conseqüência da própria definição de $U(n)$. Portanto, o conjunto $U(n)$ é um *grupo* para o produto de classes.

Para as aplicações que virão a seguir é importante determinar a ordem de $U(n)$. Começamos definindo a função ϕ que, a cada número inteiro positivo n, associa um outro inteiro positivo, a ordem de $U(n)$. Esta é a *função ϕ de Euler* ou *função totiente*. Assim a ordem de $U(n)$ é denotada por $\phi(n)$.

Vamos ver como calcular ϕ. Começamos com alguns casos especiais. Seja p um número primo, então todos os inteiros positivos menores que p são

primos com p. Logo

$$U(p) = \mathbb{Z}_p \setminus \{\overline{0}\}$$

tem $p - 1$ elementos. Portanto $\phi(p) = p - 1$.

Também é fácil calcular $\phi(p^k)$. Para isto temos que contar os inteiros positivos menores que p^k cujo máximo divisor comum com p^k é 1. Mas $\text{mdc}(a, p^k) = 1$ significa que p não divide a. Então basta contar os inteiros positivos menores que p^k que não são divisíveis por p. Na verdade é mais fácil contar aqueles que *são* divisíveis. Se $0 \leqslant a < p^k$ é divisível por p, então

$$a = p \cdot b \text{ onde } 0 \leqslant b < p^{k-1}.$$

Portanto há p^{k-1} inteiros positivos menores que p^k que *são* divisíveis por p. Logo há $p^k - p^{k-1}$ que *não são* divisíveis por p. Isto é,

$$\phi(p^k) = p^k - p^{k-1} = p^{k-1}(p - 1).$$

Para obtermos uma fórmula geral é necessário provar o seguinte resultado.

Teorema. *Se* m, n *são inteiros positivos tais que* $\text{mdc}(m, n) = 1$, *então*

$$\phi(mn) = \phi(m)\phi(n).$$

Para provar isto usamos o teorema chinês do resto, interpretado em forma de tabela, como na seção 3 do capítulo 7. Vamos recordar como se faz isto. Temos dois inteiros positivos m e n que satisfazem $\text{mdc}(m, n) = 1$. Escrevemos na horizontal os números de 0 a $m - 1$ e na vertical de 0 a $n - 1$. Assim a tabela tem $m \cdot n$ casas, cada uma endereçada por dois números a, b onde $0 \leqslant a \leqslant m - 1$ e $0 \leqslant b \leqslant n - 1$. Vamos preencher a casa a, b da tabela com o inteiro x entre 0 e $mn - 1$ que satisfaz

$$x \equiv a \pmod{m}$$
$$x \equiv b \pmod{n}.$$

De acordo com o teorema chinês do resto, cada casa é preenchida por exatamente um número entre 0 e $mn - 1$. Assim, não há casas em branco, e nenhum número aparece repetido. Usando a notação acima, vamos dizer que o número x da tabela tem *coordenadas* (a, b).

Demonstração do Teorema: Digamos que $\overline{x} \in \mathbb{Z}_{mn}$, onde $0 \leqslant x \leqslant mn - 1$. Sejam a e b as coordenadas de x na tabela. Vamos começar mostrando a seguinte afirmação:

Afirmação: $\overline{x} \in U(mn)$ se, e somente se, $\overline{a} \in U(m)$ e $\overline{b} \in U(n)$.

É fácil ver que se $\overline{x} \in U(mn)$ então $\overline{a} \in U(m)$ e $\overline{b} \in U(n)$. De fato, se $\overline{x} \in U(mn)$ tem inverso $\overline{x}' \in U(mn)$, então $xx' \equiv 1 \pmod{mn}$. Portanto $xx' - 1$ é divisível por mn. Em particular $xx' - 1$ é divisível por m; donde $xx' \equiv 1 \pmod{m}$. De $x \equiv a \pmod{m}$ concluímos que $ax' \equiv 1 \pmod{m}$; logo $\overline{a}$ é inversível em $\mathbb{Z}_m$. A demonstração de que $\overline{b} \in U(n)$ é análoga e fica a seu cargo escrevê-la.

Vejamos como provar a recíproca. Seja $\overline{x}$ um elemento de $\mathbb{Z}_{mn}$ e suponhamos que suas coordenadas satisfazem $\overline{a} \in U(m)$ e $\overline{b} \in U(n)$. Queremos mostrar que $\overline{x}$ é inversível módulo mn. Seja $\overline{a'}$ o inverso de $\overline{a}$ em $\mathbb{Z}_m$ e $\overline{b'}$ o inverso de $\overline{b}$ em $\mathbb{Z}_n$. O inverso de $\overline{x}$, se existir, deve ocupar alguma posição na tabela: quais serão suas coordenadas? É razoável esperar que sejam $\overline{a'}$ e $\overline{b'}$, conforme definidos anteriormente. Pelo teorema chinês do resto, tem que existir um inteiro y tal que $0 \leqslant y \leqslant mn - 1$ e

$$\begin{aligned} y &\equiv a' \pmod{m} \\ y &\equiv b' \pmod{n}. \end{aligned}$$

Vamos mostrar que $\overline{y} \in \mathbb{Z}_{mn}$ é o inverso de $\overline{x}$. Como $x \equiv a \pmod{m}$ e $y \equiv a' \pmod{m}$, temos que

$$xy \equiv aa' \equiv 1 \pmod{m}.$$

Logo $xy - 1$ é divisível por m. Um argumento análogo mostra que $xy - 1$ é divisível por n. Como $\text{mdc}(m, n) = 1$ podemos concluir, pelo lema da seção 6 do capítulo 2, que $xy - 1$ é divisível por mn. Isto é $\overline{x} \cdot \overline{y} = \overline{1}$ em $\mathbb{Z}_{mn}$, como queríamos mostrar. Provamos assim a afirmação feita anteriormente.

Usando a afirmação fica fácil provar o teorema. Queremos calcular $\phi(mn)$. Por definição isto é o número de elementos de $U(mn)$. Portanto, usando a afirmação anterior, o que queremos é contar o número de casas da tabela cujas coordenadas estão, respectivamente, em $U(m)$ e $U(n)$. Mas este número é igual ao produto do número de elementos de $U(m)$ pelo número de elementos de $U(n)$; que é $\phi(m)\phi(n)$. Portanto $\phi(mn) = \phi(m)\phi(n)$ e o teorema está provado.

Temos agora todos os ingredientes necessários para obter uma fórmula geral para $\phi(n)$. Dado n um inteiro positivo, comece por fatorar n,

$$n = p_1^{e_1} \dots p_k^{e_k}$$

onde $p_1 < \cdots < p_k$ são primos distintos. Pelo teorema

$$\phi(n) = \phi(p_1^{e_1}) \dots \phi(p_k^{e_k}).$$

Usando o cálculo de ϕ para potências de primos que fizemos anteriormente

$$\phi(n) = p_1^{e_1-1} \dots p_k^{e_k-1}(p_1 - 1) \dots (p_k - 1).$$

É conveniente fazer um exemplo numérico simples. Se $n = 120 = 8 \cdot 3 \cdot 5$, então,

$$\phi(120) = 2^2(2-1)(3-1)(5-1) = 32.$$

A utilidade desta fórmula geral é muito limitada. Para aplicá-la é necessário fatorar n; um problema notoriamente difícil, como já tivemos oportunidade de observar.

5. Subgrupos

Se um grupo H está dentro de um grupo G e se *compartilham da mesma operação*, dizemos que H é um *subgrupo* de G. É conveniente que sejamos mais precisos em nossa definição.

Seja G um grupo com a operação $*$. Um subconjunto não vazio H de G é um *subgrupo* de G se:
(1) Para todo $a, b \in H$ temos que $a * b \in H$.
(2) O elemento neutro de G está em H.
(3) Para todo $a \in H$, seu inverso a' também está em H.

Observe que (1) apenas afirma que $*$ (que é a operação de G) também é uma operação para H. Entre os exemplos que já vimos há vários subgrupos. Por exemplo, para a operação de soma, $\mathbb{Z}$ é um subgrupo de $\mathbb{Q}$, que é um subgrupo de $\mathbb{R}$, que é um subgrupo de $\mathbb{C}$. Para a operação de produto: os racionais não nulos são um subgrupo dos reais não nulos, que por sua vez são um subgrupo dos complexos não nulos. As matrizes de determinante igual a 1 formam um subgrupo, com a operação produto, do grupo das matrizes de determinante diferente de zero. Observe que qualquer grupo tem pelo menos dois subgrupos: o próprio grupo e o subgrupo formado apenas pelo elemento neutro.

Por outro lado, $\mathbb{Q} \setminus \{0\}$ é um grupo para o produto e está contido em $\mathbb{Q}$, que é um grupo para a soma. Mas neste caso *não* temos um subgrupo, porque os dois conjuntos *não compartilham a mesma operação*: em um deles a operação é o produto, no outro é a soma.

Os grupos finitos são muito interessantes. O fato do conjunto admitir uma operação com certas propriedades faz com que existam muitas relações numerológicas curiosas entre a ordem do grupo e as ordens de seus subgrupos. Isto torna muito mais fácil achar os subgrupos de um dado grupo finito.

Vamos estudar a mais simples destas relações, que é conhecida como *Teorema de Lagrange*.

Teorema de Lagrange. *Em um grupo finito, a ordem de qualquer subgrupo divide a ordem do grupo.*

A demonstração do teorema de Lagrange será feita na seção 8. Antes precisamos nos convencer da utilidade deste resultado para o cálculo dos subgrupos de um dado grupo. Vamos aplicar o teorema a um grupo que já conhecemos bastante, o grupo D_3 das simetrias do triângulo eqüilátero. Lembre-se que D_3 tem ordem 6. Portanto, segundo o teorema, D_3 só pode ter subgrupos de ordem 1, 2, 3 e 6, que são os divisores de 6. Como todo subgrupo tem que conter o elemento neutro, só há um subgrupo possível de ordem 1, que é $\{e\}$. Também só há um subgrupo possível de ordem 6, que é o próprio D_3. Assim precisamos apenas determinar os subgrupos de ordem 2 e 3 de D_3. Antes de prosseguir, veremos uma maneira padrão de construir subgrupos a partir de elementos de um grupo.

6. Subgrupos cíclicos

Seja G um grupo *finito* com uma operação $*$. Digamos que a é um elemento de G. Vamos escrever

$$a^k = a * a * \ldots * a \quad (k \text{ vezes}).$$

Esta é a k-ésima *potência* de a. Consideremos o conjunto das potências de a

$$H = \{e, a, a^2, a^3, \ldots\}.$$

Aparentemente trata-se de um conjunto infinito. *Aparentemente,* porque $H \subseteq G$, que é um conjunto finito. Logo H também tem que ser finito. Mas isto significa que potências muito grandes de a vão ter que ser iguais a potências menores. Assim vão existir inteiros positivos $n > m$, tais que $a^m = a^n$. Seja a' o inverso de a em G. Multiplicando esta equação por $(a')^m$, obtemos $a^{n-m} = e$, o elemento neutro. Concluímos duas coisas importantes:

(1) Dado um elemento qualquer $a \in G$ existe um inteiro positivo k tal que $a^k = e$.
(2) Se $a^k = e$, então o inverso de a é a^{k-1}, uma potência de a. Logo o inverso de a pertence a H.

Em particular temos que H é um subgrupo de G.

Qual a ordem de H? Digamos que k é o menor inteiro positivo tal que $a^k = e$. Se $n > k$, então podemos dividir n por k, obtendo $n = k \cdot q + r$, onde $0 \leqslant r \leqslant k - 1$. Então,

$$a^n = a^{kq+r} = (a^k)^q * a^r.$$

Como $a^k = e$, segue-se que $a^n = a^r$. Concluímos que não precisamos nos preocupar com as potências de a com expoente maior que $k - 1$, porque são iguais a outras potências, de expoentes menores. Logo,

$$H = \{e, a, a^2, \ldots, a^{k-1}\}.$$

Além disto, todos estes elementos são distintos. De fato, se $r \leqslant s < k$ e $a^r = a^s$, então $a^{s-r} = e$. Como $s - r < k$, que é a ordem de a, temos que $s - r = 0$; ou seja $r = s$. Portanto dois elementos do conjunto acima são iguais se, e só se, correspondem a potências com os mesmos expoentes. Assim a ordem de H é k.

Com isto temos um método simples para construir subgrupos de G. Dado um grupo finito G, escolha $a \in G$ diferente do elemento neutro e, então:

(1) O conjunto H formado pelas potências de a em G é um subgrupo.
(2) A ordem de H é igual ao menor inteiro positivo k tal que $a^k = e$.

É conveniente introduzir a seguinte terminologia. Se o subgrupo H é igual ao conjunto das potências de um elemento a, diremos que a é um *gerador* de H. O menor inteiro positivo k tal que $a^k = e$ é a *ordem* de a. Portanto, se H tem a como gerador então a ordem de H (isto é, o número de elementos de H) é igual à ordem de a. Um subgrupo que admite um elemento gerador é chamado de *cíclico*.

Uma aplicação simples do que vimos nos permite descrever a estrutura de um grupo G cuja ordem é um número primo p. Escolha um elemento qualquer $a \neq e$ de G e considere o subgrupo H formado pelas potências de a. Pelo teorema de Lagrange, a ordem de H tem que dividir a ordem de G, que é p, um número primo. Logo H tem ordem 1 ou p. Mas H contém pelo menos dois elementos, a saber e e a. Logo H não pode ter ordem 1. Resta-nos que H tenha ordem p. Assim H tem tantos elementos quantos tem G; isto é $H = G$. Isto nos permite concluir que, se G é um grupo de ordem prima, então:

(1) G é cíclico;
(2) G só tem dois subgrupos, o próprio G e $\{e\}$;
(3) cada elemento de G diferente do neutro gera todo o grupo.

Vemos assim que todos os grupos de ordem prima são cíclicos. Mas não é verdade que se um grupo é cíclico, então tem que ter ordem prima. Um

exemplo simples é $U(5)$, que tem ordem $\phi(5) = 4$ e é cíclico gerado por $\overline{2}$. Voltaremos a este exemplo no capítulo 10, quando discutirmos o teorema da raiz primitiva.

7. Determinando subgrupos

Vamos aplicar tudo isto para obter os subgrupos de D_3. Já vimos, na seção 5, que é suficiente determinar os subgrupos de ordem 2 e 3 de D_3. Vamos começar calculando as ordens dos elementos. É fácil ver que ρ e ρ^2 têm ordem 3, e que as três reflexões têm, cada uma, ordem 2. O conjunto das potências de ρ é

$$R = \{e, \rho, \rho^2\}$$

que coincide com o conjunto das potências de ρ^2. Logo R é um subgrupo cíclico de D_3 que é gerado tanto por ρ, quanto por ρ^2. As reflexões nos dão mais 3 subgrupos de D_3, cada um de ordem 2:

$$\{e, \sigma_1\},\ \{e, \sigma_2\} \text{ e } \{e, \sigma_3\}.$$

Será que D_3 tem mais algum subgrupo? Certamente que não há nenhum subgrupo cíclico além destes, porque já usamos todos os elementos de D_3 como geradores de algum subgrupo. Mas será que há algum subgrupo que não seja cíclico? Pondo a questão desta maneira, a resposta é sim. O subgrupo de D_3 que não é cíclico é D_3 mesmo. Para ser cíclico, D_3 teria que conter um elemento de ordem 6, o que não ocorre. Mas não é isto o que queríamos saber. A pergunta correta é: será que há algum subgrupo próprio de D_3 que não seja cíclico? Dizemos que é *próprio* o subgrupo que não é igual ao grupo todo.

Sabemos que os subgrupos próprios de D_3 só podem ter ordens 2 ou 3. Mas estes são números primos, e já vimos que um grupo cuja ordem é um primo *tem que ser cíclico*. Portanto D_3 não pode ter subgrupos próprios que não sejam cíclicos. Isto nos permite concluir, também, que D_3 não tem nenhum subgrupo além dos que já achamos.

Para que você não comece a achar que os subgrupos próprios de um grupo qualquer são sempre cíclicos, vejamos outro exemplo. Desta vez o grupo é

$$U(16) = \{\overline{1}, \overline{3}, \overline{5}, \overline{7}, \overline{9}, \overline{11}, \overline{13}, \overline{15}\}.$$

com a operação de multiplicação módulo n. Este grupo tem ordem 8, logo só pode ter subgrupos de ordem 1, 2, 4 ou 8. Os subgrupos de ordem 1 e 8 são $\{\overline{1}\}$ e $U(16)$, respectivamente.

Vamos determinar as ordens dos elementos de $U(16)$. Um cálculo fácil mostra que os elementos de ordem 2 são $\overline{7}$, $\overline{9}$, $\overline{15}$ e os elementos de ordem

4 são $\overline{3}$, $\overline{5}$, $\overline{11}$ e $\overline{13}$. Primeira conclusão: $U(16)$ não é cíclico porque não tem elemento de ordem 8. Além disto, este grupo tem um subgrupo próprio que não é cíclico. Isto acontece porque o conjunto

$$\{\overline{1}, \overline{7}, \overline{9}, \overline{15}\}$$

é um subgrupo de $U(16)$, mas seus elementos têm ordem menor ou igual a 2.

Note que este subgrupo tem ordem 4, um número composto. Isto não é mera coincidência. Lembre-se que já sabemos que um grupo de ordem prima *tem que ser cíclico*. Logo, em um grupo qualquer, se um subgrupo não é cíclico então sua ordem tem que ser um número composto.

Com o que vimos na seção anterior podemos facilmente obter uma generalização do teorema de Fermat para o caso em que o módulo não é primo.

Teorema de Euler. *Sejam* $n > 0$ *e* a *números inteiros. Se* $\text{mdc}(a, n) = 1$ *então*

$$a^{\phi(n)} \equiv 1 \pmod{n}.$$

A demonstração é uma conseqüência do teorema de Lagrange. Como a e n são primos entre si, $\overline{a} \in U(n)$. Logo a ordem de $\overline{a}$ divide a ordem de $U(n)$, que é $\phi(n)$. Se a ordem de $\overline{a}$ é k, isto significa que $\phi(n) = kr$, para algum inteiro positivo r. Assim,

$$(\overline{a})^{\phi(n)} = (\overline{a}^k)^r = \overline{1},$$

que é equivalente à equação do teorema de Euler.

8. Teorema de Lagrange

Começaremos recordando o enunciado do teorema.

Teorema de Lagrange. *Em um grupo finito, a ordem de qualquer subgrupo divide a ordem do grupo.*

Para provar este teorema vamos ter que usar *relações de equivalência*. Se você já esqueceu deste assunto, é melhor fazer uma revisão. Seja G um grupo e H um subgrupo de G. Definiremos uma relação de equivalência em G usando H. Sejam x e y elementos de G. Dizemos que x e y são *congruentes módulo* H, e escrevemos $x \equiv y \pmod{H}$ se

$$x * y' \in H.$$

Lembre-se que y' é o inverso de y em G.

Observe que um caso especial desta relação já é nosso conhecido. Digamos que o grupo G seja o conjunto $\mathbb{Z}$ com a soma e o subgrupo H o conjunto dos

múltiplos de um inteiro positivo n. Se x, y são inteiros, então $y' = -y$ e portanto $x \equiv y \pmod{H}$ significa que $x - y \in H$. Isto é, $x - y$ é um múltiplo de n. Portanto, neste exemplo, $x \equiv y \pmod{H}$ é a mesma coisa que $x \equiv y \pmod{n}$.

Precisamos verificar que a relação definida acima é de equivalência. Faremos isto testando cada uma das propriedades. Se $x, y, z \in G$ então teremos que:

(1) **Reflexividade:** $x * x' = e$, o elemento neutro. Como $e \in H$ então $x \equiv x \pmod{H}$.
(2) **Simetria:** se $x \equiv y \pmod{H}$ então $x * y' \in H$. Mas um subgrupo contém o inverso de cada um de seus elementos. Logo $y * x'$, que é o inverso de $x * y'$, pertence a H. Mas $y * x' \in H$ significa que $y \equiv x \pmod{H}$.
(3) **Transitividade:** Digamos que $x \equiv y \pmod{H}$ e $y \equiv z \pmod{H}$. Estas duas equações se traduzem como $x * y' \in H$ e $y * z' \in H$. Como o produto de dois elementos de H está em H, temos que $(x * y') * (y * z') \in H$. Efetuando o cálculo obtemos $x * z' \in H$; que é equivalente a dizer que $x \equiv z \pmod{H}$.

Observe que cada uma das condições necessárias para que H seja um subgrupo corresponde, exatamente, a uma das propriedades que a congruência módulo H tem que satisfazer para ser uma relação de equivalência. Assim, a reflexividade corresponde ao fato de H conter o elemento neutro; a simetria ao fato do inverso de um elemento de H também pertencer a H; a transitividade ao fato do 'produto' de dois elementos de H também pertencer a H. O teorema de Lagrange depende do delicado equilíbrio entre estas duas definições.

Agora que sabemos que a congruência módulo H é uma relação de equivalência, vamos determinar a classe de equivalência de um elemento $x \in G$ por esta relação. Por definição a classe de equivalência de x é

$$\{y \in G : y \equiv x \pmod{H}\}.$$

Mas $y \equiv x \pmod{H}$ é o mesmo que dizer que $y * x' \in H$. Isto é $y = h * x$, para algum elemento $h \in H$. Portanto a classe de equivalência de x pode ser escrita como

$$\{h * x : h \in H\}.$$

Isto sugere a notação Hx para esta classe. Observe que a classe do elemento neutro e corresponde ao próprio H. Com isto estamos em condições de provar o teorema de Lagrange.

Demonstração do teorema de Lagrange: Seja G um grupo finito e denotemos por $*$ sua operação. Seja H um subgrupo de G. Para provar o teorema precisamos apenas contar quantos elementos tem uma classe de equivalência para a congruência módulo H. Se $x \in G$, já vimos que sua classe de equivalência é

$$Hx = \{h * x : h \in H\}.$$

Precisamos mostrar que o número de elementos de Hx é *igual* ao número de elementos de H. Como Hx é obtido multiplicando um elemento fixo x de G pelos elementos de H, então Hx *não* pode ter mais elementos que H. Mas poderia ter menos elementos. Isto aconteceria se dois elementos distintos de H, quando multiplicados por x dessem resultados iguais. Felizmente isto não pode acontecer em um grupo. De fato se

$$h_1, h_2 \in H \text{ e } h_1 * x = h_2 * x$$

então multiplicando esta última equação, à direita, por x', o inverso de x, obtemos $h_1 = h_2$. Concluímos portanto que Hx e H têm o mesmo número de elementos.

Este é o ponto crucial da demonstração, o que falta é mera reunião do que já sabemos. De fato, G é a união de todas as classes de equivalência. Como duas classes ou são iguais ou disjuntas, então o número de elementos de G é igual à soma do número de elementos de cada classe–onde estamos considerando apenas classes distintas, é claro. Mas todas as classes têm o mesmo número de elementos, que é a ordem de H. Logo o número de elementos de G é igual à ordem de H vezes o número de classes. Portanto a ordem de H é um divisor da ordem de G, como queríamos verificar.

9. Exercícios

1. O grupo D_4 das simetrias do quadrado tem ordem 8.

(1) Escreva cada um dos elementos de D_4 como permutação dos vértices do quadrado.

(2) Determine o inverso de cada um dos elementos de D_4.

(3) Seja ρ a rotação de $90°$ e σ uma das reflexões do quadrado. Verifique que $\sigma\rho = \rho^3\sigma$.

(4) Escreva a tabela do grupo D_4.

2. Seja G um grupo. Mostre que se o quadrado de qualquer elemento do grupo é a identidade, então o grupo é abeliano.

3. Calcule $\phi(125)$, $\phi(16200)$ e $\phi(10!)$.

4. Seja n um inteiro positivo e p um fator primo de n.
(1) Mostre que $p - 1$ divide $\phi(n)$.
(2) Mostre que pode acontecer que p *não* divida $\phi(n)$.
(3) Mostre que $n > \phi(n)$.

5. Determine os valores de n para os quais $\phi(n) = 18$. Faça o mesmo para $\phi(n) = 10$ e $\phi(n) = 14$.

6. Mostre que se $\phi(n)$ é um número primo então $n = 3, 4$ ou 6.

7. Seja k um inteiro positivo. Os exercícios anteriores ilustram bem as dificuldades de resolver a equação $\phi(n) = k$. Curiosamente, há um algoritmo relativamente simples para resolver $n\phi(n) = k$. Seja $k = n\phi(n)$ e suponhamos que p é o maior primo que divide k. Mostre que:

(1) O maior primo que divide n é igual a p.
(2) O expoente de p na fatoração de $n\phi(n)$ tem que ser ímpar.

Concluímos de (2) que se p tem expoente par na fatoração de k, então a equação $k = n\phi(n)$ não tem solução. Suponhamos, então, que p tem expoente ímpar na fatoração de k. Supondo que n existe e $n = p^e \cdot c$, onde $\text{mdc}(c, p) = 1$, temos que

$$k = n\phi(n) = p^{2e-1}(p-1)c\phi(c).$$

Podemos usar isto para calcular e, uma vez que conhecemos o expoente de p na fatoração de k. Uma vez encontrado e, podemos escrever

$$\frac{k}{p^{2e-1}(p-1)} = c\phi(c),$$

e usar o mesmo método para achar o maior primo que divide c. Por que este algoritmo pára?

8. Mostre que, se n é um inteiro positivo que satisfaz $\phi(n) = n - 1$, então n é primo.

9. Escrevendo $n = 2^k r$, onde r é um número ímpar, mostre que se $\phi(n) = n/2$ então n é uma potência de 2.

10. Mostre que se m divide n então $\phi(mn) = m\phi(n)$.

11. Determine todos os subgrupos do grupo D_4 das simetrias do quadrado.

12. Verifique que $U(2)$ e $U(4)$ são grupos cíclicos, mas que $U(8)$ não é cíclico.

13. Suponhamos que G é um grupo *cíclico* finito de ordem n. Mostre que se m divide n então G tem um elemento de ordem m.

14. Considere o grupo $U(20)$.
(1) Determine a ordem de $U(20)$.
(2) Determine a ordem de cada elemento de $U(20)$.
(3) Verifique que o grupo $U(20)$ não é cíclico.
(4) Determine os subgrupos de ordem 4 de $U(20)$.
(5) Determine um subgrupo de $U(20)$ que não seja cíclico.

15. Seja G um grupo e S_1 e S_2 dois subgrupos de G. Mostre que:
(1) $S_1 \cap S_2$ é um subgrupo de G;
(2) se as ordens de S_1 e S_2 são primas entre si, então $S_1 \cap S_2 = \{e\}$;
(3) $S_1 \cup S_2$ pode não ser um subgrupo de G.

Sugestão: Para mostrar (2) lembre-se que $S_1 \cap S_2$ é subgrupo de S_1 e de S_2; use o teorema de Lagrange e o fato das ordens serem primas entre si para mostrar que $S_1 \cap S_2$ só pode ter um elemento. Para verificar (3) basta dar um exemplo em que a união de subgrupos não é um grupo; experimente $G = D_3$.

16. Seja n um inteiro positivo, composto e ímpar. Considere o seguinte subconjunto de $U(n)$:

$$H(n) = \{\overline{b} \in U(n) : n \text{ é pseudoprimo para a base } b\}.$$

Determine se cada uma das afirmações abaixo é verdadeira ou falsa, *justificando cuidadosamente suas respostas*.
(1) $H(n)$ é um subgrupo de $U(n)$.
(2) $H(n)$ não pode ser igual a $U(n)$ porque n é composto.
(3) $U(n)$ não pode ter um elemento de ordem $n-1$ porque n é composto.

17. Calcule a forma reduzida de 7^{9876} módulo 60 e de 3^{87654} módulo 125.

18. Seja $p > 0$ um número primo e r um inteiro positivo. Use o teorema de Euler, aplicado a p^r para mostrar que p^r é um pseudoprimo para uma base b, prima com p, se, e somente se, $b^{p-1} \equiv 1 \pmod{p^r}$.

19. Use o exercício anterior para mostrar que 1093^2 é um pseudoprimo para a base 2.

20. Escreva um programa, baseado no exercício 18, para determinar os pseudoprimos para a base 2 da forma p^2, onde p é um primo menor ou igual a $r > 0$. Isto nos dá uma versão melhorada do programa proposto no exercício 11 do capítulo 6.

21. Escreva um programa que, tendo por entrada um inteiro $k > 0$, determina $\phi(k)$. O programa vai consistir, essencialmente, de um algoritmo para fatorar k completamente. Adapte o programe de modo que encontre todos os valores de k menores que 10^5 tais que $\phi(k) = \phi(k+1)$. Não se sabe se existem infinitos inteiros positivos k satisfazendo esta propriedade.

22. Dizemos que um inteiro $k > 0$ é um *totiente* se a equação $\phi(n) = k$ admite alguma solução. Escreva um programa que implemente o algoritmo descrito no exercício 7. Quais os números menores que 10^5 que você consegue verificar que são totientes usando este algoritmo?

9

Mersenne e Fermat

Neste capítulo estudamos de maneira sistemática os números de Mersenne e Fermat. Entre outras coisas obteremos métodos para calcular fatores primos destes números. Para isto será necessário usar as ferramentas que desenvolvemos anteriormente, sobretudo os resultados da teoria de grupos.

1. Números de Mersenne

A maneira mais fecunda de produzir primos 'grandes' é através de fórmulas exponenciais. As fórmulas exponenciais para primos mais antigas estão associadas aos nomes de Fermat e Mersenne, matemáticos do século XVII. A idéia é procurar primos entre os números da forma $2^n \pm 1$. Existem razões históricas para considerar primos desta forma, como vimos no capítulo 3.

Consideremos primeiramente os *números de Mersenne*. Seja n um inteiro positivo. O n-ésimo número de Mersenne é

$$M(n) = 2^n - 1.$$

Já vimos que se n é composto, então $M(n)$ é composto. De fato, se $n = rs$, então

$$2^n - 1 = (2^r)^s - 1 = (2^r - 1)(2^{r(s-1)} + 2^{r(s-2)} + \cdots + 2^r + 1).$$

Portanto $M(r)$ é fator de $M(n) = M(rs)$. Naturalmente, $M(s)$ também é um fator de $M(n)$.

Assim, se o que desejamos é obter primos entre os números de Mersenne, devemos procurá-los entre os números $M(p)$, com p primo. Entretanto o fato de p ser primo não é suficiente para garantir que $M(p)$ também seja primo, como vimos no capítulo 3. O que vamos descrever a seguir é um método, devido a Fermat, que nos permite encontrar fatores para números de Mersenne não muito grandes, de maneira relativamente eficiente. A idéia é descobrir uma fórmula geral para os números primos que podem ser fatores

de um número de Mersenne. Antes de descrever o método precisamos de um resultado complementar ao teorema de Lagrange.

Lema chave. *Digamos que* G *é um grupo finito munido de uma operação* $*$. *Seja* $a \in G$. *Um inteiro positivo* t *satisfaz* $a^t = e$ *se, e somente se,* t *é divisível pela ordem de* a.

A demonstração é simples. Vamos chamar a ordem de a de s. Se s divide t então $t = sr$, para algum inteiro positivo r, e

$$a^t = (a^s)^r = e.$$

A recíproca é menos imediata. Suponhamos que $a^t = e$. Como a ordem é o *menor* inteiro positivo s tal que $a^s = e$, então $s \leqslant t$. Dividindo t por s, temos

$$t = sq + r \text{ onde } 0 \leqslant r < s.$$

Assim,

$$e = a^t = (a^s)^q * a^r = a^r,$$

já que $a^s = e$. Como $r < s$ isto só pode acontecer se $r = 0$, ou teríamos uma contradição com o fato de s ser a ordem de a.

Voltando aos números de Mersenne, digamos que $p \neq 2$ é um número primo e que q é um fator primo de $M(p) = 2^p - 1$. Então

$$2^p \equiv 1 \pmod{q}.$$

Podemos ler esta equação como uma identidade no grupo $U(q) = \mathbb{Z}_q \setminus \{\overline{0}\}$, a saber

$$\overline{2}^p = \overline{1}.$$

Qual a ordem de $\overline{2}$ em $U(q)$? Como vimos no lema chave, a ordem de $\overline{2}$ deve dividir p. Mas p é primo; logo $\overline{2}$ tem ordem 1 ou p. Como $p \neq 2$, por hipótese, $\overline{2}$ tem ordem p.

Por outro lado, pelo teorema de Fermat

$$\overline{2}^{q-1} = \overline{1} \text{ em } U(q).$$

Novamente pelo lema chave, a ordem de 2 (que é p) divide $q-1$. Isto é, existe um inteiro k tal que $q - 1 = kp$.

Podemos ser ainda mais precisos. De fato, $M(p) = 2^p - 1$ é um número ímpar, logo todo fator de $M(p)$ é ímpar. Em particular, q é ímpar. Portanto $q - 1$ é par. Como p também é ímpar, concluímos que k, na fórmula acima,

tem que ser par. Assim, $q - 1 = 2rp$ para algum inteiro r. Provamos o seguinte resultado.

Método de Fermat. *Seja* $p \neq 2$ *um primo e* q *um fator primo de* $M(p)$. *Então* $q = 1 + 2rp$ *para algum inteiro positivo* r.

Vamos usar este método para achar um fator de $M(11) = 2047$. Temos que qualquer fator *primo* de $M(11)$ é da forma $q = 1 + 22r$. Vamos atribuir valores a r e testar em cada caso se temos um fator ou não. Antes de começar é bom ter uma estimativa de quantos valores de r vamos ter que considerar *no pior caso possível*. Sabemos que se $M(p)$ for composto então tem um fator $\leqslant \sqrt{M(p)}$. Logo

$$\sqrt{M(p)} \geqslant q = 1 + 2rp.$$

Como

$$\sqrt{M(p)} < 2^{p/2},$$

então

$$r < \frac{2^{p/2} - 1}{2p}$$

Quando $p = 11$ isto dá $r < 2$. Só temos que tentar $r = 1$–não poderia ser mais eficiente. De fato, fazendo $r = 1$, temos $q = 23$, que é fator de 2047. O outro fator primo é $89 = 1 + 44 \cdot 2$. É interessante comparar com o que aconteceria se estivéssemos usando o método usual, que consiste em tentar achar um fator *entre todos os primos menores que* $\sqrt{M(p)}$. Neste caso $[\sqrt{2047}] = 45$. Assim teríamos que testar todos os primos ímpares até 45; e há 13 tais primos.

A história dos números de Mersenne é uma mina de excentricidades. Uma das melhores é a história de F.N. Cole que, em um encontro da American Mathematical Society em 1903 deu uma palestra silenciosa, em que verificou, fazendo a conta, que

$$M(67) = 193707721 \cdot 761838257287.$$

A audiência aplaudiu entusiasticamente! Até hoje não se conhece uma maneira simples de chegar a esta fatoração. Em 1992 um supercomputador Cray-2 foi usado para mostrar que $M(756839)$ é primo. O maior primo de Mersenne conhecido é $M(1398269)$.

2. Números de Fermat

Vamos considerar agora os números da forma 2^n+1. Desejamos saber quando estes números podem ser primos. Suponhamos que $p = 2^n + 1$ é primo. Então,

$$\overline{2}^n = -\overline{1} \text{ em } U(p). \tag{2.1}$$

Portanto

$$\overline{2}^{2n} = \overline{1} \text{ em } U(p).$$

Logo, a ordem de $\overline{2}$ em $U(p)$ divide $2n$. Precisamos calculá-la exatamente. Observe que, por causa da equação (2.1), a ordem de $\overline{2}$ *não* pode ser n nem um divisor de n. Como a ordem divide $2n$, temos então que é um mútiplo de 2. Digamos que a ordem de $\overline{2}$ é $2r$, onde r é um inteiro positivo que divide n. Assim $\overline{2}^{2r} = \overline{1}$ em $U(p)$. Daí

$$\overline{0} = \overline{2}^{2r} - \overline{1} = (\overline{2}^r - \overline{1})(\overline{2}^r + \overline{1})$$

em $\mathbb{Z}_p$. Como estamos supondo que p é primo, podemos concluir que

$$2^r \equiv 1 \pmod{p} \quad \text{ou} \quad 2^r \equiv -1 \pmod{p}.$$

Isto é p divide $2^r + 1$ ou $2^r - 1$. Mas $p = 2^n + 1$ e $n > r$, o que nos dá uma contradição. Logo $r = n$ e a ordem de $\overline{2}$ em $U(p)$ é $2n$.

Como $p - 1 = 2^n$ temos, pelo teorema de Fermat, que

$$\overline{2}^{2^n} = \overline{1} \text{ em } U(p).$$

Logo a ordem de $\overline{2}$ (que é $2n$) divide 2^n. Em particular, concluímos que n é uma potência de 2. Isto é, $2^n + 1$ *só pode ser primo quando* n *é uma potência de* 2.

Por isso, estaremos apenas interessados nos números da forma $2^{2^k} + 1$. Estes são os *números de Fermat*, denotados por $F(k)$. Em uma carta de 1640, endereçada ao cavaleiro Frenicle, Fermat sugeriu que estes números são sempre primos. De fato $F(k)$ é primo para k variando de 0 a 4. Mas $F(5)$ é composto. Isto foi provado pelo matemático suíço L. Euler em 1730. Ironicamente, o método adotado por Euler copia o método inventado por Fermat para achar fatores de números de Mersenne. Vejamos como fazer isto.

Digamos que q é um fator *primo* de $F(k)$. Então

$$\overline{2}^{2^k} = -\overline{1} \text{ em } U(q). \tag{2.2}$$

Segue do lema chave que a ordem de $\overline{2}$ divide 2^{k+1}; mas a equação (2.2) nos diz também que a ordem não pode ser uma potência menor que 2^{k+1}. Logo

a ordem de $\overline{2}$ em $U(q)$ é exatamente 2^{k+1}. Mas, pelo teorema de Fermat, a ordem de $\overline{2}$ divide $q-1$, assim $q-1=2^{k+1}r$.

Método de Euler. *Se* q *é um fator primo de* $F(k)$ *então existe um número inteiro positivo* r *tal que* $q=1+2^{k+1}r$.

Vamos determinar um fator de $F(5)=2^{32}+1$. Pelo método de Euler, um fator *primo* de $F(5)$ tem que ser da forma $q=1+64r$. É necessário verificar a divisibilidade por q para os valores de r para os quais

$$q<\sqrt{2^{32}+1}\leqslant 66000.$$

Isto dá $r<1031$, um número surpreendentemente grande. O menor valor de r para o qual q é primo é $r=3$, que dá $q=193$. Fazendo a conta

$$2^{32}\equiv(2^8)^4\equiv 63^4\equiv 108 \pmod{193}.$$

Portanto 193 não é fator de $F(5)$. Para $r=4$ temos $q=257$ que é primo, mas

$$2^{32}\equiv 1 \pmod{257};$$

logo 257 também não é fator. O valor seguinte de r que produz um primo é $r=7$, para o qual $q=449$ mas

$$2^{32}\equiv(2^{16})^2\equiv 431^2\equiv 324 \pmod{449};$$

logo 449 não é fator. O próximo primo ocorre com $r=9$, para o qual $q=577$. Neste caso

$$2^{32}\equiv 287 \pmod{577}$$

de modo que 577 também não é fator de $F(5)$. Finalmente, quando $r=10$, temos $q=641$, que é um fator de $F(5)$. Observe que *por sorte* o fator é pequeno e podemos encontrá-lo usando o método de Euler e uma máquina de calcular. Infelizmente nem sempre temos tanta sorte. Como $F(k)$ cresce muito rapidamente (já que é duplamente exponencial), temos geralmente que testar um número tão grande de valores de r para achar um fator que o método de Euler torna-se rapidamente inútil.

Se o que você deseja é fatorar um determinado número de Fermat, então existem métodos muito mais eficientes que o método de Euler. Talvez mais surpreendentemente, existe uma maneira muito eficaz de determinar se $F(k)$ é primo ou não sem tentar fatorar. Estudaremos este método no capítulo seguinte. Sabe-se bastante sobre os números de Fermat. Por exemplo, são conhecidas fatorações completas em primos para todos os números de Fermat com $k\leqslant 9$ e $k=11$. Além disso pelo menos um fator primo é conhecido

para cada um dos números de Fermat para os quais $k \leqslant 32$, exceto para $k = 14, 20, 22, 24, 28$ e 31. Na verdade, os *únicos* números de Fermat primos conhecidos têm $k \leqslant 4$ e datam do século XVII. Acredita-se que estes são os únicos números de Fermat que são primos.

Por que tanto interesse nos números de Fermat? Há várias razões. O *pedigree* destes números é uma delas. Eles têm uma longa história, e muitos matemáticos importantes já trabalharam no problema. Além disto, são uma boa fonte de exemplos de números grandes e difíceis de fatorar, ótimos para testar um novo algoritmo de fatoração. Por sua vez, estes algoritmos exigem a execução de uma quantidade enorme de operações aritméticas e lógicas simples. Isto os torna ideais para detectar possíveis erros de lógica ou aritmética nos chamados supercomputadores. Por exemplo, o Cray X-MP foi testado desta maneira.

Existe um outra razão pela qual estes números são interessantes. Em 1801 Gauss mostrou que, para que um polígono regular de n lados possa ser construído com *régua e compasso*, é preciso que n seja o produto de uma potência de 2 por números de Fermat *primos*. Assim um polígono de 17 lados pode ser construído com régua e compasso, por que $17 = F(2)$; mas um heptágono não pode, porque 7 não é um primo de Fermat. Para mais detalhes veja [**Artin 1991, Cap. 13, seção 4**].

3. Fermat, novamente

O maior número de Fermat para o qual se conhece um fator é $F(23471)$; o fator é $5 \cdot 2^{23473} + 1$. Este é, também, o maior número de Fermat que se sabe ser composto. Trata-se de um número enorme, e você deve estar imaginando que deve haver excelentes métodos de fatoração para números de Fermat, para chegar a um tal resultado. Para tornar o mistério ainda mais denso, só lhe resta saber que o método utilizado neste caso foi o método inventado por Euler, que estudamos na seção 2. Impossível? Nem tanto.

A idéia é muito simples. Começamos inventando um número que possa ser fator de um número de Fermat. Para isto basta escolher um inteiro ímpar k e um inteiro qualquer n e construir o número $q = k \cdot 2^n + 1$. De acordo com o método de Euler, este número pode ser um divisor de $F(m)$, desde que $m \leqslant n-1$. Dizer que q divide $F(m)$ equivale a dizer que a seguinte congruência é verificada

$$2^{2^m} \equiv -1 \pmod{q}.$$

Para trabalhar com números tão grandes, precisamos de uma maneira eficiente de calcular estas congruências. Como só entram em jogo potências

de 2, isto pode ser feito com certa facilidade. A observação básica é que

$$(2^{2^i})^2 = 2^{2^{i+1}}.$$

Procedemos da seguinte maneira. Inicializamos $r = 2^{2^5}$ e $i = 5$. A variável i apenas contabiliza o tamanho do expoente. O programa, então, substitui r pelo resto de r^2 por q e incrementa i de 1. Se $r = q - 1$, então q divide $F(i)$. Se não então repetimos o processo, até que $i = n - 1$. Observe que q não pode dividir $F(i)$ quando $i \geqslant n$, como vimos na seção 2.

Este método (com algumas melhoras) foi usado recentemente para achar fatores para $F(15)$, $F(25)$, $F(27)$ e $F(147)$. Nenhum destes números havia sido fatorado anteriormente. Os cálculos foram realizados nos computadores da Convex, uma empresa americana, por G.B. Gostin, e foram publicados em 1995; veja [**Gostin 1995**].

O algoritmo usado foi escrito em C e *assembler*, com algumas rotinas 'paralelizadas'. O programa começa gerando um milhão de valores possíveis de q a serem testados. Em seguida são eliminados aqueles valores de q que são divisíveis por primos pequenos. Os valores de q que sobreviverem ao crivo são então testados nas congruências.

Reproduzimos abaixo uma parte da tabela que publicada em [**Gostin 1995, p. 394**]. A tabela contém apenas os valores de m, k e n, onde $k \cdot 2^n + 1$ é fator de 2^m.

m	k	n
15	17753925353	17
64	17853639	67
353	18908555	355
885	16578999	887
1082	82165	1084
1225	79707	1231
1451	13143	1454
3506	501	3508
6390	303	6393
6909	6021	6912

4. O teste de Lucas-Lehmer

Nesta seção, apresentamos um teste de primalidade para números de Mersenne que é muito eficiente. Concebido originalmente por E. Lucas (o inventor das torres de Hanói) em 1878, este teste foi melhorado por D. H. Lehmer em 1932, e é atualmente conhecido como *teste de Lucas-Lehmer*.

O principal ingrediente do teste de Lucas-Lehmer é a seqüência de inteiros positivos $S_0, S_1, S_2, \ldots$, definida recursivamente por

$$S_0 = 4 \quad \text{e} \quad S_{k+1} = S_k^2 - 2.$$

Mostraremos, em primeiro lugar, que os inteiros desta seqüência podem ser escritos como potências de números irracionais.

Seja $\omega = 2 + \sqrt{3}$ e $\varpi = 2 - \sqrt{3}$. Provaremos por indução em n que

$$\omega^{2^n} + \varpi^{2^n} = S_n. \tag{4.1}$$

É claro que $\omega + \varpi = S_0$. Suponha, então, que $\omega^{2^{n-1}} + \varpi^{2^{n-1}} = S_{n-1}$. Elevando ambos os membros ao quadrado, obtemos

$$\omega^{2^n} + 2(\omega\varpi)^{2^{n-1}} + \varpi^{2^n} = S_{n-1}^2.$$

Como $\omega\varpi = 1$, segue que

$$\omega^{2^n} + \varpi^{2^n} = S_{n-1}^2 - 2,$$

que é igual a S_n por definição.

Teste de Lucas-Lehmer. *Seja p um primo positivo. O número de Mersenne $M(p)$ é primo se, e somente se,* $S_{p-2} \equiv 0 \pmod{M(p)}$.

Provaremos apenas que a condição é necessária; a demonstração de que é também suficiente está além das possibilidades de um livro elementar como este. A demonstração que apresentaremos apareceu originalmente em [**Bruce 1993**] e, embora elementar, utiliza números irracionais de uma maneira que é difícil de justificar heuristicamente. Contudo, uma vez que você esteja disposto a aceitar o aparecimento destes números irracionais, o resto da prova segue de perto o padrão das demonstrações das seções anteriores. Para entender de onde vêm os números irracionais consulte [**Bressoud 1989, capítulos 10 e 11**].

A demonstração do teste de Lucas-Lehmer será expressa na linguagem da teoria de grupos, mas o grupo a ser utilizado é bem mais exótico que $U(p)$. O ponto de partida é o subconjunto $\mathbb{Z}[\sqrt{3}]$ constituído pelos números da forma $a + b\sqrt{3}$, onde $a, b \in \mathbb{Z}$. Esses são número reais, de modo que podem ser somados e multiplicados. A soma e o produto de dois números em $\mathbb{Z}[\sqrt{3}]$ também pertencem a $\mathbb{Z}[\sqrt{3}]$. Além disso, tal qual $\mathbb{Z}$, o conjunto $\mathbb{Z}[\sqrt{3}]$ é um grupo com respeito à adição, mas não com respeito à multiplicação. Tudo isto é facilmente verificado. Note que cada $a \in \mathbb{Z}$ pode ser escrito na forma $a = a + 0\sqrt{3}$, de modo que $\mathbb{Z} \subseteq \mathbb{Z}[\sqrt{3}]$.

Seja agora q um inteiro primo, e escreva

$$I(q) = \{q\alpha : \alpha \in \mathbb{Z}[\sqrt{3}]\}.$$

É claro que $0 = 0q \in I(q)$. Como $q\alpha + q\beta = q(\alpha + \beta)$, concluímos que a soma de dois números em $I(q)$ também está em $I(q)$. Além disso, para cada $\alpha \in \mathbb{Z}[\sqrt{3}]$, tanto $q\alpha$, quanto $-q\alpha$, pertencem a $I(q)$. Assim $I(q)$ é um subgrupo do grupo aditivo $\mathbb{Z}[\sqrt{3}]$.

Desta forma a relação *congruência módulo* $I(q)$ é uma relação de equivalência em $\mathbb{Z}[\sqrt{3}]$, como foi mostrado na seção 8 do capítulo 8. Lembre-se que, se $\alpha, \beta \in \mathbb{Z}[\sqrt{3}]$, então $\alpha \equiv \beta \pmod{I(q)}$ quando $\alpha - \beta \in I(q)$.

Agora, se $\alpha \in \mathbb{Z}[\sqrt{3}]$, então $\alpha = a_1 + a_2\sqrt{3}$, onde $a_1, a_2 \in \mathbb{Z}$. Dividindo a_1 e a_2 por q, obtemos $a_1 = qb_1 + r_1$ e $a_2 = qb_2 + r_2$, com $0 \leqslant r_1, r_2 < q$. Escrevendo $\rho = r_1 + r_2\sqrt{3}$, temos que

$$\alpha - \rho = q(b_1 + b_2\sqrt{3}).$$

Portanto, $\alpha \equiv \rho \pmod{I(q)}$. O número ρ é chamado de *forma reduzida* de α módulo $I(q)$. Como o resto da divisão de dois inteiros é única, cada elemento de $\mathbb{Z}[\sqrt{3}]$ tem apenas uma forma reduzida módulo $I(q)$. Note que há exatamente q^2 formas reduzidas distintas módulo $I(q)$ em $\mathbb{Z}[\sqrt{3}]$.

Por outro lado, cada classe de equivalência de $\mathbb{Z}[\sqrt{3}]$ módulo $I(q)$ pode ser representada por um elemento em forma reduzida. Além disso, duas classes representadas por elementos cujas formas reduzidas são diferentes precisam ser distintas. Assim o conjunto $\mathbb{Z}_q[\sqrt{3}]$, das classes de equivalência módulo $I(q)$, tem q^2 elementos.

A classe de equivalência de $\alpha \in \mathbb{Z}[\sqrt{3}]$ módulo $I(q)$ será denotada por $\widetilde{\alpha}$. Podemos definir uma multiplicação em $\mathbb{Z}_q[\sqrt{3}]$ por

$$\widetilde{\alpha}\widetilde{\beta} = \widetilde{\alpha\beta}.$$

A demonstração de que esta definição é independente da escolha dos representantes das classes segue de perto a demonstração do resultado correspondente para aritmética modular na seção 3 do capítulo 4. Se $\alpha = a_1 + a_2\sqrt{3}$ e $\beta = b_1 + b_2\sqrt{3}$, então um cálculo elementar mostra que $\widetilde{\alpha}\widetilde{\beta}$ está representada por

$$(a_1b_1 + 3a_2b_2) + (a_1b_2 + a_2b_1)\sqrt{3}.$$

É fácil verificar que esta multiplicação é associativa, comutativa, e tem $\widetilde{1}$ como elemento neutro. Contudo, $\mathbb{Z}_q[\sqrt{3}]$ não é um grupo para esta operação;

por exemplo, $\widetilde{0}$ não tem inverso em $\mathbb{Z}_q[\sqrt{3}]$. Como fizemos para a aritmética modular, contornamos esta dificuldade considerando o conjunto $V(q)$ formado pelos elementos inversíveis de $\mathbb{Z}_q[\sqrt{3}]$. Este conjunto é um grupo, porque o produto de elementos inversíveis de $\mathbb{Z}_q[\sqrt{3}]$ tem como resultado um elemento que também é inversível: basta copiar a demonstração do resultado análogo para $U(q)$ que se encontra na seção 4 do capítulo 8. Note que, como

$$V(q) \subset \mathbb{Z}_q[\sqrt{3}] \setminus \{\widetilde{0}\},$$

temos que a ordem de $V(q)$ é necessariamente menor que q^2. Além disso, como $\omega\varpi = 1$, concluímos que, tanto ω, quanto ϖ, pertencem a $V(q)$. Estamos agora prontos para provar o teste de Lucas-Lehmer.

Demonstração do teste de Lucas-Lehmer: Suponha que, para algum primo p, o número de Mersenne $M(p)$ divide S_{p-2}. Por (4.1), existe um inteiro r tal que

$$\omega^{2^{p-2}} + \varpi^{2^{p-2}} = rM(p).$$

Multiplicando esta equação por $\omega^{2^{p-2}}$, e relembrando que $\omega\varpi = 1$, obtemos

$$\omega^{2^{p-1}} + 1 = rM(p)\omega^{2^{p-2}}. \tag{4.2}$$

Suponha, agora, que $M(p)$ é composto e que q é seu menor fator primo, e vamos tentar chegar a uma contradição. Como q divide $M(p)$, obtemos a partir de (4.2) que

$$\widetilde{\omega}^{2^{p-1}} = -\widetilde{1}, \tag{4.3}$$

em $\mathbb{Z}_q[\sqrt{3}]$. Elevando ambos os membros de (4.3) ao quadrado,

$$\widetilde{\omega}^{2^p} = \widetilde{1},$$

Segue disto, e do lema chave, que a ordem de $\widetilde{\omega}$ tem que dividir 2^p; mas a equação (4.3) também nos diz que esta ordem não pode ser uma potência de 2 menor que 2^p. Portanto, a ordem de $\widetilde{\omega}$ em $V(q)$ é 2^p. Mas, pelo teorema de Lagrange, a ordem de $\widetilde{\omega}$ divide a ordem de $V(q)$. Como $V(q)$ tem ordem menor ou igual a $q^2 - 1$, concluímos que $2^p \leqslant q^2 - 1$. Entretanto, q é o menor divisor primo de $M(p)$, de modo que $q^2 \leqslant M(p)$. Assim,

$$2^p \leqslant q^2 - 1 < 2^p - 1$$

que é a contradição desejada. Com isso, se $M(p)$ divide S_{p-2} então $M(p)$ é primo. Isto mostra que a condição é necessária. Para uma demonstração de que também é suficiente, veja [**Bressoud 1989, Teorema 11.10, p. 175**].

Embora este teste seja um pouco mais difícil de provar, ele é muito fácil de implementar e de usar. Em 1978, dois estudantes secundários americanos, Laura Nickel and Curt Noll, usaram o teste de Lucas-Lehmer no computador da universidade local para mostrar que $M(21701)$ é primo. Seu feito rendeu uma nota na primeira página do New York Times! Atualmente o teste de Lucas-Lehmer é a base do GIMPS (*Great Internet Mersenne Prime Search*). Para obter os programas do GIMPS visite (http://www.mersenne.org/) que oferece programas de graça (com código fonte) através da *world wide web* de modo que qualquer pessoa que tenha um computador em casa pode se juntar à busca por primos de Mersenne gigantes. O maior primo de Mersenne conhecido é $M(6972593)$ e foi descoberto por Nayan Hajratwala em junho de 1999 usando os programas do GIMPS. Este primo tem 2.098.960 algarismos, e é o primeiro primo conhecido com mais de um milhão de algarismos!

5. Exercícios

1. Sejam p, q números primos. Mostre que se a equação $x^p \equiv 1 \pmod{q}$ tem uma solução $x \not\equiv 1 \pmod{q}$ então $q \equiv 1 \pmod{p}$.

2. Determine as soluções da equação $x^{17} \equiv 1 \pmod{43}$.

3. Determine os geradores de $U(17)$. Use isto para resolver a equação $7^x \equiv 6 \pmod{17}$.

4. Use o método de Fermat para achar fatores primos dos números de Mersenne $M(11)$ e $M(29)$ e para mostrar que $M(7)$ é primo.

5. Use o método de Euler para mostrar que $F(4)$ é um número primo.

6. Seja $k \geqslant 2$ um inteiro e $\alpha = 2^{2^{k-2}}(2^{2^{k-1}} - 1)$. Seja p um divisor primo de $F(k) = 2^{2^k} + 1$. Mostre que:
(1) $\alpha^2 \equiv 2 \pmod{p}$.
(2) α tem ordem 2^{k+2} módulo p.
(3) $p = 2^{k+2}r + 1$, onde r é um inteiro positivo.

Observe que que este resultado torna a busca de fatores primos pelo método de Euler um pouco mais eficiente.

7. Ache um número de Fermat $F(k)$ que tenha $7 \cdot 2^{14} + 1$ como fator primo.

8. Em 1640, o cavalheiro Frenicle perguntou a Fermat, através de Mersenne, se existiam números perfeitos entre 10^{20} e 10^{22}. Sabemos que Frenicle estava se referindo a números perfeitos *pares* ou *euclidianos*; veja Cap. 2, exercícios

8, 9 e 10. O objetivo desta questão é dar a demonstração de Fermat de que estes números não existem. Lembre-se que mostramos nos exercícios mencionados acima que todo número perfeito par é da forma $2^{n-1}(2^n - 1)$, onde $2^n - 1$ é primo.

(1) Mostre que se $n \geqslant 2$ então $-1 < \log(1 - 2^{-n}) < 0$, onde log denota logaritmos decimais. Conclua disto que

$$n \log 2 - 1 < \log(2^n - 1) < n \log 2.$$

(2) Aplicando logaritmos decimais à desigualdade

$$10^{20} < 2^{n-1}(2^n - 1) < 10^{22}$$

e usando o resultado de (1), mostre que $35 \leqslant n \leqslant 37$. Lembre-se que n é inteiro.

(3) Mostre que $M(n)$ nunca é primo se $35 \leqslant n \leqslant 37$.
Para mais detalhes consulte **[Weil 1987, Cap. II, §IV]**.

9. Programe o método de Fermat para achar fatores de um número de Mersenne. O programa deve ter como entrada um primo $p > 0$ e como saída o menor fator primo de $M(p) = 2^p - 1$ ou uma mensagem indicando que $M(p)$ é primo. O programa vai consistir, basicamente, de uma implementação do algoritmo apresentado na seção 2 do apêndice para calcular 2^n módulo q, onde q é da forma $q = r \cdot 2p + 1$ com $0 \leqslant r \leqslant [(2^{p/2} - 1)/2p]$. Se a forma reduzida de 2^p módulo q for 1 então q é fator de $M(p)$; se nenhum tal q for encontrado, então $M(p)$ é primo. Observe que *não* vale a pena testar se q é primo antes de verificar se é fator de $M(p)$, porque isto só tornaria o programa mais lento. Use este programa para determinar quais os primos p entre 2 e 257 para os quais $M(p)$ é primo; cf. Cap. 3, seção 2.

10. Programe o algoritmo apresentado na seção 3. O algoritmo deve ter como entrada o primo p e deve procurar o valor de m (se existir) para o qual p divide $F(m)$. Observe que, escrevendo p na forma $p = k \cdot 2^n + 1$, temos que $m < n$, o que nos dá um limite além do qual não adianta continuar a busca. Use este programa para achar o número de Fermat que é divisível por $37 \cdot 2^{16} + 1$ e o que é divisível por $11131 \cdot 2^{12} + 1$. Observe que não seria possível determinar estes fatores usando um PC se precisássemos determinar todos os algarismos dos números de Fermat correspondentes.

10

Raízes primitivas

Neste capítulo vamos mostrar que $U(p)$ é um grupo cíclico quando p é primo. Veremos que isto tem importantes conseqüências, além de servir de inspiração para um teste que permite provar que um número é primo sem tentar achar fatores. É exatamente com este teste que começaremos o capítulo.

1. Teste de Lucas

Suponhamos que desejamos determinar se um dado número inteiro positivo, ímpar, n é primo. Uma estratégia possível é tentar mostrar que $U(n)$ tem ordem $n-1$; isto é, que $\phi(n) = n-1$. Isto significa que se $a < n$ é um inteiro positivo, então $\text{mdc}(a, n) = 1$; o que só é possível se n for primo. À primeira vista esta não parece ser uma boa idéia. Afinal, para achar $\phi(n)$ sem fatorar n precisaríamos contar quantos elementos tem $U(n)$. Esta tarefa é claramente impossível se n for muito grande, mesmo com a ajuda de um computador. Felizmente, existe uma maneira de contar os elementos de $U(n)$ sem achá-los todos. Tudo depende do seguinte teorema, que só provaremos na seção 5.

Teorema da raiz primitiva. *Se p é primo então $U(p)$ é cíclico.*

Segundo o teorema, se p é primo então existe um elemento $\overline{b} \in U(p)$, cuja ordem é exatamente $p-1$, ou seja,

$$\overline{b}^{p-1} = \overline{1} \text{ mas } \overline{b}^{r} \neq \overline{1}$$

se $r < p-1$. Isto sugere a seguinte estratégia. Digamos que é dado um número ímpar n e que achamos $\overline{b} \in U(n)$ cuja ordem é exatamente $n-1$. Pelo teorema de Lagrange, a ordem de $\overline{b}$ tem que dividir a ordem de $U(n)$. Logo $n-1$ tem que dividir $\phi(n)$. Entretanto $\phi(n) \leqslant n-1$; donde concluímos que $\phi(n) = n-1$. Portanto n é primo. De acordo com o *teorema da raiz primitiva* um tal b sempre existirá se n for realmente primo. Encontrá-lo,

porém, é uma questão de sorte.

Para aplicar esta estratégia à determinação da primalidade de um número ímpar n precisamos apenas de uma maneira eficiente de mostrar que a ordem de um elemento de $U(n)$ é exatamente $n-1$.

Teste de Lucas. *Seja n um inteiro positivo ímpar e b um inteiro tal que $2 \leqslant b \leqslant n-1$. Se*

$$b^{n-1} \equiv 1 \pmod{n} \quad e \quad b^{(n-1)/p} \not\equiv 1 \pmod{n},$$

para cada fator primo p de $n-1$, então n é primo.

Demonstração: Seja k a ordem de $\overline{b}$ em $U(n)$. Queremos mostrar que $k = n-1$. Como $\overline{b}^{n-1} = \overline{1}$, sabemos, pelo lema chave, que k divide $n-1$. Digamos que $n-1 = k \cdot t$. Queremos mostrar que $t = 1$. Suponhamos, por absurdo, que $t > 1$. Então t é divisível por algum primo q. Mas se q divide t, então q divide $n-1$. Portanto $(n-1)/q$ e t/q são inteiros. Além disto, como

$$\frac{n-1}{q} = k \cdot \frac{t}{q}$$

concluímos que k divide $(n-1)/q$. Aplicando novamente o lema chave, deduzimos que

$$\overline{b}^{(n-1)/q} = \overline{1};$$

o que contradiz as hipóteses. Portanto $t = 1$ e, conseqüentemente, $k = n-1$. Pelo teorema de Lagrange, a ordem de $\overline{b}$ tem que dividir a ordem de $U(n)$. Logo $n-1$ divide $\phi(n) \leqslant n-1$. Assim $\phi(n) = n-1$ e n tem que ser primo.

Os testes anteriores detectavam *com certeza* apenas números compostos. Este teste determina *com certeza* se um dado número é primo. Mas observe que, para aplicá-lo com sucesso, precisamos ser capazes de fatorar $n-1$. De fato, a segunda equação do teste tem que ser verificada para *cada fator primo de* $n-1$. Na prática há muitos primos interessantes para os quais esta condição é facilmente verificada. Um exemplo são os fatores primos dos números de Fermat. Além disso, não há um método eficiente para escolher a base b. Nada disto é muito surpreendente, afinal o teste de Miller padecia do mesmo mal.

Este teste é excelente para números de Fermat, já que $F(k)-1$ tem apenas um fator primo, que é 2. Para estes números temos o seguinte resultado devido a Jean François Théophile Pepin (1826-1904).

Teste de Pepin. *O número de Fermat* $F(k)$ *é primo, para* $k > 1$ *se, e somente se,*

$$5^{(F(k)-1)/2} \equiv -1 \pmod{F(k)}.$$

Observe que se a equação acima vale, então $F(k)$ é certamente primo, pelo teste de Lucas. A recíproca é muito mais difícil de verificar, e depende de um famoso teorema da teoria de números, a *lei de reciprocidade quadrática*; veja [**Hardy e Wright 1994, Cap. VI**], por exemplo. Por isso não vamos prová-la aqui.

Como um exemplo, aplicaremos o Teste de Pepin ao número de Fermat $F(4)$. Temos que

$$\frac{F(4)-1}{2} = 2^{15},$$

mas

$$5^{2^{15}} \equiv 5^{32768} \equiv 65536 \equiv -1 \pmod{F(4)}.$$

Logo $F(4)$ passa no teste; portanto é primo. Os cálculos necessários para aplicar o teste de Pepin são fáceis de realizar porque envolvem apenas calcular o quadrado de números módulo $F(k)$, e isto se faz rapidamente.

Como um segundo exemplo, vamos mostrar que

$$R(19) = \underbrace{1111111111111111111}_{19}.$$

é primo. Para poder aplicar o teste de Lucas precisamos primeiro achar a fatoração completa de $R(19) - 1$. Fazendo isto, obtemos

$$R(19) - 1 = 2 \cdot 3^2 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 19 \cdot 37 \cdot 52579 \cdot 333667.$$

Nossa primeira escolha de base será $b = 2$. Usando um sistema de computação algébrica, é fácil verificar que

$$2^{R(19)-1} \equiv 1 \pmod{R(19)}.$$

Mas, infelizmente,

$$2^{(R(19)-1)/2} \equiv 1 \pmod{R(19)}.$$

Assim a condição (2) do teste de Lucas falha para $b = 2$ e $p = 2$. Portanto, 2 não é uma boa escolha de base.

Em seguida escolhemos $b = 3$. Usando novamente um sistema de computação algébrica, descobrimos que

$$3^{R(19)-1} \equiv 1 \pmod{R(19)};$$

de modo que a condição (1) do teste de Lucas vale. Precisamos agora calcular as formas reduzidas de

$$3^{(R(19)-1)/p}$$

módulo $R(19)$ para cada um dos primos p que aparecem na fatoração de $R(19)-1$. Este é o conteúdo da seguinte tabela.

Fator primo p	Resto de $3^{(R(19)-1)/p}$ por $R(19)$
2	$R(19)-1$
3	933000903779960656
5	97919522321038174
7	742392324159673027
11	920873402557886628
13	114592042672083983
19	10^{11}
37	397724716798816350
52579	760105763664485871
333667	555602369615218524

Assim a condição (2) do teste de Lucas vale para cada um dos fatores primos de $R(19)-1$. Portanto, $R(19)$ é, de fato, um número primo.

Só se conhecem 5 primos da forma $R(n)$. O primeiro é $R(2) = 11$, e o segundo é $R(19)$. Os outro são $R(23)$, $R(317)$ e $R(1031)$. Não é difícil provar que $R(23)$ é primo usando uma versão melhorada do teste de Lucas que vamos descrever na próxima seção.

2. Outro teste determinístico de primalidade

Façamos um exemplo, bem simples, para ilustrar uma das dificuldades na aplicação do teste de Lucas. Digamos que queremos usá-lo para verificar que $n = 41$ é primo. Começamos fatorando $n-1 = 40$, obtendo $n-1 = 2^3 \cdot 5$. Portanto precisamos achar um inteiro b tal que $2 \leqslant b \leqslant 40$ e que também satisfaça a cada uma das equações abaixo:

$$b^{40} \equiv 1 \pmod{41}$$
$$b^{20} \not\equiv 1 \pmod{41}$$
$$b^{8} \not\equiv 1 \pmod{41}.$$

Começamos com $b = 2$, mas logo verificamos que $2^{20} \equiv 1 \pmod{41}$. Passa-

mos a $b = 3$, mas embora $3^{20} \equiv 40 \pmod{41}$, temos $3^8 \equiv 1 \pmod{41}$. Para tornar as coisas mais exasperantes, $2^8 \equiv 10 \pmod{41}$. Neste caso, bases diferentes satisfazem as equações que advêm de fatores diferentes de $n-1$. Infelizmente o teste requer que uma única base satisfaça a todas as equações. Para 41 a menor destas bases é 7.

Em 1975 Brilhart, Lehmer e Selfridge observaram que não é necessário usar uma base única no teste de Lucas. Podemos nos dar ao luxo de escolher uma base distinta para cada primo. Com isto obtemos um teste um pouco mais eficiente.

Teste de Primalidade. *Seja* $n > 0$ *um inteiro tal que*

$$n - 1 = p_1^{e_1} \dots p_r^{e_r}$$

onde $p_1 < \cdots < p_r$ *são primos. Se, para cada* $i = 1, \dots, r$ *existirem inteiros positivos* b_i *(*$2 \leqslant b_i \leqslant n-1$*) que satisfaçam*

$$b_i^{n-1} \equiv 1 \pmod{n}$$
$$b_i^{(n-1)/p_i} \not\equiv 1 \pmod{n},$$

então n *é primo.*

Não é necessário que os b_i's sejam todos distintos. Vejamos porque estas equações forçam a que n seja primo.

Vamos considerar o caso $i = 1$, o mesmo se aplica a $i = 2, \dots, r$. Em primeiro lugar precisamos calcular a ordem s_1 de b_1 em $U(n)$. Segue do lema chave e da equação $b_1^{n-1} \equiv 1 \pmod{n}$ que s_1 divide $n-1$. Logo os primos que aparecem na fatoração de s_1 estão entre os primos $p_1, \dots, p_r$. Assim

$$s_1 = p_1^{k_1} \dots p_r^{k_r}$$

onde $k_1 \leqslant e_1, \dots, k_r \leqslant e_r$.

Por outro lado, sabemos que $b_1^{(n-1)/p_1} \not\equiv 1 \pmod{n}$. Isto significa que $(n-1)/p_1$ *não* é divisível por s_1. Mas

$$(n-1)/p_1 = p_1^{e_1-1} p_2^{e_2} \dots p_r^{e_r}.$$

Comparando com a fatoração de s_1 vemos que a única maneira de s_1 não dividir $(n-1)/p_1$ é que $k_1 = e_1$. Em outras palavras, $p_1^{e_1}$ divide s_1.

Não podemos perder de vista que s_1 é a ordem de b_1 em $U(n)$. Pelo teorema de Lagrange, s_1 divide a ordem de $U(n)$; isto é, s_1 divide $\phi(n)$. Logo $p_1^{e_1}$ divide s_1 que divide $\phi(n)$: portanto $p_1^{e_1}$ divide $\phi(n)$.

Como as contas acima se aplicam também a $i = 2, \ldots, r$ concluímos que as equações do teste implicam que $p_1^{e_1}, p_2^{e_2}, \ldots, p_r^{e_r}$ dividem $\phi(n)$. Mas estes números são potências de primos distintos, logo primos entre si. Assim, pelo lema da seção 6 do capítulo 2, o produto

$$p_1^{e_1} \ldots p_r^{e_r} = n - 1$$

também divide $\phi(n)$. Como $\phi(n) \leqslant n - 1$ resta apenas que $\phi(n) = n - 1$, e isto é suficiente para garantir que n é primo.

Finalmente vamos reunir tudo o que aprendemos numa estratégia para verificar primalidade. Suponhamos que temos um número ímpar n, muito grande, e queremos saber se é primo:

(1) Verificamos, por tentativa, se n é divisível pelos primos até 5000.
(2) Supondo que n não é divisível por estes primos, obtivemos um estoque de bases para aplicar o teste de Miller.
(3) Supondo que n passou no teste de Miller para todas estas bases, tentamos finalmente aplicar o teste descrito acima.

3. Números de Carmichael

Já vimos que é possível caracterizar todos os números de Carmichael. Chamamos este resultado de teorema de Korselt. Antes de provar o *teorema da raiz primitiva*, vamos usá-lo para completar a demonstração do teorema de Korselt. Começamos relembrando seu enunciado.

Teorema de Korselt. *Um inteiro positivo impar n é um número de Cármichael se, e somente se, cada fator primo p de n satisfaz as duas condições seguintes:*
(1) p^2 não divide n;
(2) $p - 1$ divide $n - 1$.

No Capítulo 6, seção 2, provamos que se (1) e (2) são satisfeitas para um número n, então n é um número de Carmichael. Reciprocamente, mostramos que se n é um número de Carmichael, então (1) é satisfeita. Falta, apenas, verificar que, se n é um número de Carmichael, então (2) é satisfeita. Para isto usaremos o *teorema da raiz primitiva*.

Suponhamos que n é um número de Carmichael. Isto significa que $b^n \equiv b \pmod{n}$ para todo inteiro b que satisfaz $1 < b < n - 1$. Seja p um primo que divide n. Pelo *teorema da raiz primitiva* o grupo $U(p)$ é cíclico, gerado por uma classe $\overline{a}$.

Mas $a^n - a$ é divisível por n. Como p divide n, temos também que p

divide $a^n - a$. Assim $a^n \equiv a \pmod{p}$. Como p é primo e não divide a, concluímos que a é inversível módulo p, e portanto, que $a^{n-1} \equiv 1 \pmod{p}$. Pelo lema chave, isto implica que a ordem de $\overline{a}$ divide $n-1$. Mas $\overline{a}$ é um gerador de $U(p)$, logo sua ordem é $p-1$. Assim $p-1$ divide $n-1$, que é o que nos faltava verificar.

4. Preliminares

Antes de descrever a demonstração do teorema da raiz primitiva, precisamos de um resultado preliminar. Como este resultado vale para qualquer grupo abeliano, e não apenas para $U(p)$, vamos descrevê-lo neste contexto mais geral.

Lema. *Seja* G *um grupo abeliano finito munido de uma operação* $*$. *Se* G *tem elementos de ordens* r *e* s, *então* G *tem um elemento cuja ordem é igual ao mínimo múltiplo comum entre* r *e* s.

Usaremos este resultado para provar o teorema da raiz primitiva na próxima seção. A demonstração do teorema nos dará um algoritmo para achar um gerador do grupo cíclico $U(p)$. Por isso, a demonstração do lema acima é tão importante quanto o seu enunciado. Precisamos não apenas saber que o elemento existe, mas teremos que aprender a construí-lo.

Digamos que a e b são elementos de G de ordens r e s, respectivamente. Começaremos fatorando r e s em fatores primos. Digamos que

$$r = p_1^{e_1} \dots p_k^{e_k} \quad \text{e} \quad s = p_1^{f_1} \dots p_k^{f_k}.$$

onde $p_1, \dots, p_k$ são primos *distintos*. Observe que escrevemos os mesmos primos nas duas fatorações. Isto não significa que estamos supondo que r e s têm os mesmos fatores primos. Se um dos primos p_1, $p_2 \dots$ ou p_k não é fator primo de r, por exemplo, então o seu expoente na fatoração será zero.

Note também que não dissemos que os primos estão enumerados em ordem crescente. Na verdade, é preferível arranjar os primos de modo que, para um inteiro g, onde $1 \leqslant g \leqslant k$, tenhamos

$$e_1 \geqslant f_1, \dots, e_g \geqslant f_g, \quad \text{mas} \quad e_{g+1} < f_{g+1}, \dots, e_k < f_k.$$

Assim os expoentes dos g primeiros primos na fatoração de r são maiores ou iguais aos expoentes correspondentes na fatoração de s. Os expoentes dos primos restantes são menores na fatoração de r que na de s.

Escreveremos

$$r' = p_1^{e_1} \dots p_g^{e_g} \quad \text{e} \quad s' = p_{g+1}^{f_{g+1}} \dots p_k^{f_k}.$$

Observe que os fatores primos de r' e s' são distintos; portanto $\mathrm{mdc}(r', s') = 1$. Por outro lado $r's'$ é o produto de potências dos primos $p_1, \ldots, p_k$. Além disso o primo p_i aparece com um expoente que é o maior valor entre e_i e f_i. Portanto $r's'$ é igual ao mínimo múltiplo comum entre r e s.

Como r' divide r e s' divide s, existem inteiros positivos u e v tais que $r = r'u$ e $s = s'v$. Queremos mostrar que se $a \in G$ tem ordem r e se $b \in G$ tem ordem s então $c = a^u * b^v$ é um elemento de G cuja ordem é igual ao mínimo múltiplo comum entre r e s.

Antes de começar a calcular, observe que uma das hipóteses do lema é que o grupo é abeliano. Esta hipótese é necessária porque precisaremos usar que se $x, y \in G$ então

$$(x * y)^q = x^q * y^q,$$

e isto não é verdade se $*$ não for comutativa. Na verdade o resultado do lema é falso se o grupo não for abeliano. Daremos um exemplo ao final da demonstração para ilustrar isto.

Em primeiro lugar seja m o mínimo múltiplo comum entre r e s. Então r divide m e s divide m. Digamos que $m = rt = sq$ para inteiros positivos t e q. Então

$$c^m = (a^u * b^v)^m = a^{um} * b^{vm} = (a^r)^{ut} * (b^s)^{vq} = e.$$

Logo a ordem de c divide o mínimo múltiplo comum entre r e s pelo lema chave.

Por outro lado, se n é a ordem de c, então

$$e = c^n = a^{un} * b^{vn}.$$

Elevando esta equação ao expoente r', temos que

$$e = (a^{r'u})^n * b^{vnr'} = b^{vnr'},$$

já que $r'u = r$ é a ordem de a. Concluímos, usando novamente o lema chave, que a ordem de b (que é s) divide vnr'. Mas $s = s'v$; logo se s divide vnr', então s' divide nr'. Contudo r' e s' são primos entre si, por construção. Portanto, s' tem que dividir n, pelo lema da seção 6 do capítulo 2.

Um cálculo análogo, trocando os papéis de a e b e de r e s, mostra que r' divide n. Como r' e s' são primos entre si, temos, pelo lema da seção 6 do capítulo 2, que $r's'$ divide n. Mas $r's' = m$, é o mínimo múltiplo comum entre r e s. Logo m divide n. Como já havíamos visto que n divide m, obtemos $m = n$ e c é o elemento desejado.

Note que, *se* r *e* s *são primos entre si,* podemos escolher $r' = r$ e $s' = s$. Assim $u = v = 1$ e conseqüentemente $c = a * b$ tem ordem rs. Em geral, conhecendo as fatorações de r e s em primos e conhecendo a e b podemos encontrar c.

Finalmente vale a pena mostrar que é necessário supor que G seja abeliano, porque o resultado do lema pode ser falso quando o grupo não é abeliano. Daremos um exemplo no grupo D_3. Lembre-se que este é o grupo das simetrias do triângulo eqüilátero, e não é abeliano. Considere a reflexão σ_1 e a rotação ρ. A ordem de σ_1 é 2 e a ordem de ρ é 3. Se o lema valesse, então D_3 deveria ter um elemento de ordem 6, o que não é verdade. Observe que se ignorarmos o fato do grupo não ser abeliano e escrevermos assim mesmo o elemento que foi construído na demonstração do lema para este caso particular, obteremos $\sigma_1\rho = \sigma_3$ que tem ordem 2.

5. Teorema da raiz primitiva

Seja $p > 3$ um número primo. A ordem de $U(p)$ é $\phi(p) = p - 1$, um número par e composto; mesmo assim o grupo $U(p)$ é cíclico. Este resultado é conhecido como *teorema da raiz primitiva*. Uma *raiz primitiva* é a mesma coisa que um gerador de $U(p)$.

O nome raiz primitiva pode parecer peculiar, mas é fácil justificá-lo. Pelo teorema de Fermat todos os elementos de $U(p)$ são raízes da equação $x^{p-1} - \overline{1} = \overline{0}$ em $\mathbb{Z}_p$. Um gerador de $U(p)$ é uma raiz primitiva porque obtemos as outras raízes da equação como potências da raiz primitiva. Quando resolvemos equações com coeficientes complexos também nos deparamos com o mesmo fenômeno. Por exemplo, a equação $x^p - 1 = 0$ tem raiz primitiva $\cos(2\pi/p) + i\operatorname{sen}(2\pi/p)$.

A demonstração que daremos do teorema da raiz primitiva é construtiva; isto é, a partir dela pode-se escrever um algoritmo para determinar geradores de $U(p)$. Tanto o teorema, quanto a demonstração que vamos apresentar, são devidos a C. F. Gauss e apareceram em suas *Disquisitiones Arithmeticæ*, veja [**Gauss 1986,** §§73 **e** 74].

Teorema da raiz primitiva. *Se* p *é primo então o grupo* $U(p)$ *é cíclico.*

Demonstração: Podemos supor que $p \geqslant 5$, já que o teorema é obviamente verdadeiro quando $p = 2$ ou 3. Escolha um elemento qualquer $\overline{a_1} \in U(p)$, onde $1 < a_1 < p - 1$. Podemos sempre começar escolhendo $a_1 = 2$. Seja k_1 a ordem de $\overline{a_1}$. Se $k_1 = p - 1$, então já encontramos um elemento que gera $U(p)$. Logo $U(p)$ é cíclico.

Suponha, então, que $k_1 < p - 1$. Temos que $\overline{a_1}$ é uma das soluções em $\mathbb{Z}_p$ da equação $x^{k_1} - 1$. Como p é primo, esta equação não pode ter mais de k_1 soluções distintas pelo teorema da seção 4 do Cap. 5. Por outro lado, os elementos de

$$H = \{\overline{1}, \overline{a_1}, \overline{a_1}^2, \ldots \overline{a_1}^{k_1 - 1}\}$$

são soluções de $x^{k_1} - 1$. Como H contém k_1 elementos distintos, então tem que conter todas as soluções de $x^{k_1} - 1$. Mas $k_1 < p - 1$, logo existe um elemento $\overline{b} \in U(p)$ que não pertence a H. Em particular $\overline{b}$ não é uma solução $x^{k_1} - 1$. Usando o lema chave podemos concluir disto que a ordem de $\overline{b}$ não pode dividir k_1.

Seja r a ordem de $\overline{b}$. Há dois casos a considerar. Se $r = p - 1$, então $\overline{b}$ é um gerador de $U(p)$, e obtivemos o resultado desejado. Seja, então, $r < p - 1$. Pelo lema da seção 3, concluímos que existe um elemento $\overline{a_2}$ que tem ordem k_2 igual ao mínimo múltiplo comum entre k_1 e r.

Como r não divide k_1, temos que $k_2 > k_1$. Continuando assim, ou obtemos um elemento de ordem $p - 1$, como queremos; ou construímos uma seqüência estritamente crescente de inteiros positivos $k_1 < k_2 < k_3 < \ldots$ onde cada um destes números é a ordem de um elemento de $U(p)$. Em particular cada inteiro desta seqüência divide $p - 1$; portanto tem que ser menor ou igual a $p - 1$. Logo a seqüência tem que parar, produzindo assim o gerador desejado para $U(p)$.

Este método produz, de maneira sistemática, um gerador para $U(p)$, mas nem sempre produz o *menor* gerador. Note que a recíproca do teorema é falsa; por exemplo, $U(4)$ é cíclico. Portanto não basta que $U(n)$ seja cíclico para que n seja primo. Na verdade, $U(n)$ é cíclico se, e somente se, n é igual a 1, 2, 4, p^k ou $2p^k$, onde p é um primo ímpar qualquer. Para uma demonstração deste resultado veja [**Giblin 1993, Cap. 8**].

6. Calculando ordens

Nesta seção vamos determinar uma raiz primitiva de $U(41)$ usando o método de Gauss discutido acima. Antes de começar, precisamos obter um critério eficiente para calcular a ordem de um elemento de $U(p)$, já que este é um dos ingredientes essenciais do método de Gauss.

Seja p um primo ímpar e $\overline{a} \in U(p)$. Desejamos determinar a ordem de $\overline{a}$. Vamos chamá-la de k. Sabemos, pelo teorema de Lagrange, que k divide a ordem de $U(p)$. Portanto, k divide $\phi(p) = p - 1$. Para aplicar o método de

Gauss, precisamos fatorar $p-1$. Digamos que

$$p-1 = q_1^{e_1} \dots q_m^{e_m},$$

onde $q_1 < \cdots < q_m$ são primos e $e_1, \dots, e_m$ são inteiros *positivos*. Como k divide $p-1$, existem inteiros não negativos $r_1, \dots, r_m$ tais que

$$k = q_1^{r_1} \dots q_m^{r_m}$$

e $0 \leqslant r_1 \leqslant e_1, \dots, 0 \leqslant r_m \leqslant e_m$.

Assim, para achar k basta determinar os valores de $r_1, \dots r_m$. Vejamos como obter r_1, os outros expoentes são obtidos pelo mesmo método. Isto é, trabalhamos com um fator primo de cada vez. Para isto, calculamos a seqüência

$$a^{p-1}, a^{(p-1)/q_1}, a^{(p-1)/q_1^2}, \dots, a^{(p-1)/q_1^{e_1}}$$

módulo p. Observe que, pelo teorema de Fermat, o primeiro termo da seqüência é sempre 1. Digamos, então, que w é o *maior* inteiro não negativo tal que

$$a^{(p-1)/q_1^w} \equiv 1 \pmod{p}. \tag{6.1}$$

Então ou $w = e_1$, ou

$$a^{(p-1)/q_1^{w+1}} \not\equiv 1 \pmod{p}. \tag{6.2}$$

Concluímos do lema chave e da equação (6.1) que k divide $(p-1)/q_1^w$. Por outro lado, segundo a equação (6.2), k *não* divide $(p-1)/q_1^{w+1}$.

Em outras palavras, $k = q_1^{r_1} \dots q_m^{r_m}$ divide

$$q_1^{e_1-w} q_2^{e_2} \dots q_m^{e_m} = (p-1)/q_1^w,$$

mas não divide

$$q_1^{e_1-w-1} q_2^{e_2} \dots q_m^{e_m} = (p-1)/q_1^{w+1}.$$

Isto só pode acontecer se $r_1 = e_1 - w$, o que nos dá uma maneira algorítmica de achar r_1.

Passemos ao exemplo. O menor elemento diferente do elemento neutro em $U(41)$ é $\overline{2}$. Vamos usá-lo para iniciar a aplicação do método de Gauss. Antes de mais nada precisamos achar a ordem de $\overline{2}$. Usaremos o algoritmo descrito acima. Em primeiro lugar fatoramos a ordem de $U(41)$, obtendo

$$\phi(41) = 40 = 2^3 \cdot 5.$$

Calculamos

$$\overline{2}^{40/2} = \overline{2}^{20} = \overline{1}.$$

e também

$$\overline{2}^{40/2^2} = \overline{2}^{10} = \overline{40} \neq \overline{1}.$$

Logo o expoente de 2 na ordem de $\overline{2}$ é $3 - 1 = 2$. Passemos ao primo 5. Verificamos imediatamente que

$$\overline{2}^{40/5} = \overline{2}^{8} = \overline{10} \neq \overline{1}.$$

Logo, o expoente de 5 na ordem de $\overline{2}$ é $1 - 0 = 1$. Portanto $\overline{2}$ tem ordem $2^2 \cdot 5 = 20$. Em particular $\overline{2}$ não é uma raiz primitiva de $U(41)$.

Passamos ao elemento seguinte de $U(41)$, que é $\overline{3}$. Novamente precisamos calcular a ordem, temos que

$$\overline{3}^{40/2} = \overline{40} \neq \overline{1}$$
$$\overline{3}^{40/5} = \overline{1}$$

Da primeira equação concluímos que 8 divide a ordem de $\overline{3}$; da segunda, que 5 não divide a ordem de $\overline{3}$. Logo $\overline{3}$ tem ordem 8.

Para aplicar o método de Gauss, ainda precisamos fatorar $r = 20$ (que é a ordem de $\overline{2}$) e $s = 8$ (que é a ordem de $\overline{3}$) na forma discutida na seção 4. Como $r = 2^2 \cdot 5$ e $s = 2^3$, podemos tomar $r' = 5$ e $s' = 2^3$. Isto nos dá, na notação da seção 4, $u = 2^2$ e $v = 1$. Construímos assim o elemento

$$c = \overline{2}^{u} \cdot \overline{3}^{v} = \overline{2}^{4} \cdot \overline{3}^{1} = \overline{7}$$

de $U(41)$ que tem ordem $r's' = 40$. Portanto, sem nenhum cálculo adicional, podemos concluir que $\overline{7}$ é uma raiz primitiva de $U(41)$.

7. Exercícios

1. Use o teste da primalidade para verificar que 991 é um número primo.

2. Mostre que se n é um número ímpar positivo e se 4 não divide $n - 1$ então $(n-1)^{(n-1)/2} \equiv 1 \pmod{n}$.

3. Use o teste determinístico para verificar que $M(7) = 2^7 - 1$ é primo.

Lembrete: Seja $p = 2^7 - 1$ então $2^7 \equiv 1 \pmod{p}$.

4. Seja p um número primo e $n = 2p + 1$. Suponha que $2^{n-1} \equiv 1 \pmod{n}$ e n não é divisível por 3.
(1) Mostre que se q é um fator primo de n então 4 tem ordem p em $U(q)$.
(2) Mostre que q é da forma $q = kp + 1$ para algum inteiro positivo k.

(3) Mostre que como $q < n$, então $k = 1$.
(4) Use isto para concluir que n tem que ser primo.

Sugestão para (1): $2^{n-1} = 4^p$. Logo, pela hipótese, $4^p \equiv 1 \pmod{n}$. Se q é um fator primo de n então esta congruência também vale módulo q.

5. Mostre que:
(1) Se b é um número ímpar e $k \geqslant 3$ é inteiro, então $b^{2^{k-2}} \equiv 1 \pmod{2^k}$.

Sugestão: use indução em k começando com $k = 3$.
(2) $U(2^k)$ *não* é cíclico quando $k \geqslant 3$.

6. O objetivo desta questão é descrever um outro teste determinístico de primalidade, baseado no *teorema de Wilson* descrito em (3) abaixo.
(1) Seja G um grupo abeliano finito. Mostre que o produto de todos os elementos de G é igual ao produto dos elementos de G cuja ordem é 2.
(2) Seja p um primo. Mostre que os únicos elementos de ordem 2 de $U(p)$ são $\overline{1}$ e $\overline{-1} = \overline{p-1}$.
(3) Use (1) e (2) para mostrar que $(p-1)! \equiv -1 \pmod{p}$. Este resultado é conhecido como *teorema de Wilson*.
(4) Mostre que se n for composto então $(n-1)! \equiv 0 \pmod{n}$.
(5) Juntando (3) e (4) obtemos o seguinte teste de primalidade. É dado um inteiro positivo n. Então n é primo se, e somente se, $(n-1)! \equiv -1 \pmod{n}$. Qual a principal desvantagem deste teste?

7. Use o método de Gauss para achar um gerador de $U(73)$. Este exemplo é calculado por Gauss nas *Disquisitiones*; veja [**Gauss 1986,** §**74**].

8. Seja p um número primo ímpar.
(1) Mostre que se a é ímpar e $\overline{a}$ gera $U(p)$ então a classe de a em $U(2p)$ é um gerador deste grupo.
(2) Mostre que se a é par e $\overline{a}$ gera $U(p)$, então a classe de $a + p$ em $U(2p)$ é um gerador deste grupo.
(3) Conclua que $U(2p)$ é cíclico.

9. Considere um grupo cíclico finito G de ordem n e gerador g. Seja k um inteiro positivo.
(1) Mostre que g^k também é um gerador de G se, e somente se, $\text{mdc}(k, n) = 1$.
(2) Use (1) para mostrar que G tem $\phi(n)$ geradores.
(3) Quantos são os geradores de $U(p)$ quando p é primo?

10. Programe o teste de Pepin. A entrada deve ser o expoente $n \geqslant 0$; a saída deve ser uma mensagem indicando se $F(n)$ é primo ou composto. Observe

que o programa vai consistir, essencialmente, de uma implementação do algoritmo para o cálculo de potências apresentado na seção 2 do apêndice. Qual o maior número de Fermat que você consegue testar com o seu programa?

11. Escreva um programa que implemente o teste de primalidade descrito no exercício 6. O programa vai consistir, essencialmente, de um algoritmo para calcular a forma reduzida de $(n-1)!$ módulo n. Se você tentar calcular $(n-1)!$, e só depois reduzi-lo módulo n, o programa vai ficar limitado a valores muito pequenos de n. Para evitar isto, efetue as reduções módulo n à medida que for multiplicando os números entre 2 e $n-1$. Aplique o programa ao maior primo, menor que 10^k, quando $k = 1, \ldots, 6$. Quanto tempo o programa leva para verificar que cada um destes números é primo? Tente descobrir uma regra que permita relacionar (aproximadamente) o tempo decorrido ao valor de k. Use esta regra para calcular quanto tempo o programa possivelmente levaria para verificar que um número de 100 algarismos é primo.

11

Criptografia RSA

É chegada a hora de reunir tudo o que fizemos anteriormente, na descrição do método RSA. Além de entender o funcionamento do método, precisamos considerar, em mais detalhes, a sua segurança. Isto é, por que é tão difícil quebrar uma mensagem codificada no RSA?

1. Pré-codificação

A primeira coisa a fazer se desejamos usar o método RSA é converter a mensagem em uma seqüência de números. Suporemos, para simplificar, que a mensagem original é um texto onde não há números, apenas palavras. Portanto, em última análise a mensagem é constituída pelas letras que formam as palavras e pelos espaços entre palavras. Chamaremos esta primeira etapa de *pré-codificação*, para distingui-la do processo de codificação propriamente dito.

Na pré-codificação convertemos as letras em números usando a seguinte tabela de conversão:

A	B	C	D	E	F	G	H	I	J	K	L	M
10	11	12	13	14	15	16	17	18	19	20	21	22

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
23	24	25	26	27	28	29	30	31	32	33	34	35

O espaço entre duas palavras será substituído pelo número 99, quando for feita a conversão. Por exemplo, a frase *Paraty é linda* é convertida no número

$$2510271029349914992118231310.$$

Observe que a vantagem de fazer cada letra corresponder a um número de *dois* algarismos é que isto evita ambigüidades. O que aconteceria se fizéssemos A corresponder ao número 1, B ao 2, e assim por diante? Neste

caso 12 poderia ser AB ou L, que é a décima segunda letra do alfabeto, e não teríamos como decidir qual a escolha correta.

Antes de continuar precisamos determinar os parâmetros do sistema RSA que vamos usar. Estes *parâmetros* são dois primos *distintos*, que vamos denotar por p e q. Ponha $n = pq$. A última fase do processo de pré-codificação consiste em quebrar em blocos o longo número produzido anteriormente. Estes blocos devem ser números *menores* que n. Por exemplo, se escolhermos $p = 11$ e $q = 13$, então $n = 143$. Neste caso, a mensagem, cuja conversão numérica foi feita acima, pode ser quebrada nos seguintes blocos:

$$25 - 102 - 7 - 102 - 93 - 49 - 91 - 49 - 92 - 118 - 23 - 13 - 10.$$

A maneira de escolher os blocos não é única, mas certos cuidados devem ser tomados. Por exemplo, é necessário evitar que o bloco comece por 0 porque isto traria problemas na hora de decodificar. É por isso que, no exemplo acima, não escolhemos os blocos

$$25 - 102 - 71 - 029 - \ldots.$$

Observe também que os blocos em que a mensagem foi quebrada *não* correspondem a nenhuma unidade linguística – seja palavra, letra ou qualquer outra. Isto é muito bom, porque torna a decodificação por contagem de freqüência essencialmente impossível.

2. Codificando e decodificando

Encerramos assim a pré-codificação, e podemos passar à etapa de *codificação* propriamente dita. Para codificar a mensagem precisamos de n, que é o produto dos primos, e de um inteiro positivo e que seja inversível módulo $\phi(n)$. Em outras palavras, $\text{mdc}(e, \phi(n)) = 1$. Note que é fácil calcular $\phi(n)$ se conhecemos p e q; de fato,

$$\phi(n) = (p-1)(q-1).$$

Chamaremos o par (n, e) de *chave de codificação* do sistema RSA que estamos usando. Tendo submetido a mensagem à pré-codificação, temos uma seqüência de números ou blocos. Codificaremos cada bloco separadamente, e a mensagem codificada será a seqüência dos blocos codificados. Isto é muito importante: os blocos já codificados não poderão ser reunidos de modo a formar um longo número. Se isto for feito, será impossível decodificar a mensagem! Logo veremos porque.

Digamos, então, que a chave de codificação é (n, e). Como faremos para codificar um bloco b? Lembre-se que b é um inteiro positivo menor que n.

Vamos denotar o bloco codificado por $C(b)$. A receita para calcular $C(b)$ é a seguinte:

$$C(b) = \text{ resto da divisão de } b^e \text{ por } n.$$

Observe que, em termos de aritmética modular, $C(b)$ é a forma reduzida de b^e módulo n.

Vejamos o que aconteceria no exemplo que estamos considerando. Temos $p = 11$ e $q = 13$, logo $n = 143$ e $\phi(n) = 120$. Ainda precisamos escolher e. Neste exemplo, o menor valor possível para e é 7, que é o menor primo que não divide 120. Assim, o bloco 102 da mensagem anterior é codificado como o resto da divisão de 102^7 por 143. Fazendo as contas, obtemos $C(102) = 119$. É claro que executamos a conta calculando a forma reduzida de 102^7 módulo 143:

$$102^7 \equiv (-41)^7 \equiv -41^7 \equiv -81 \cdot 138 \equiv -24 \equiv 119 \pmod{143}.$$

Codificando toda a mensagem, obtemos a seguinte seqüência de blocos:

$$64 - 119 - 6 - 119 - 102 - 36 - 130 - 36 - 27 - 79 - 23 - 117 - 10$$

Vejamos como fazer para decodificar um bloco da mensagem codificada. A informação que precisamos para poder decodificar consiste de dois números: n e o inverso de e em $\phi(n)$, que denotaremos por d. Chamaremos o par (n, d) de *chave de decodificação*. Seja a um bloco da mensagem codificada, então $D(a)$ será o resultado do processo de decodificação. A receita para calcular $D(a)$ é a seguinte:

$$D(a) = \text{resto da divisão de } a^d \text{ por } n.$$

Observe que, em termos de aritmética modular, $D(a)$ é a forma reduzida de a^d módulo n.

Alguns comentários são necessários antes de fazermos um exemplo. Em primeiro lugar, é muito fácil calcular d, desde que $\phi(n)$ e e sejam conhecidos: basta aplicar o algoritmo euclidiano estendido. Por outro lado, é claro que se b é um bloco da mensagem original, então esperamos que $D(C(b)) = b$. Em outras palavras, decodificando um bloco da mensagem codificada, esperamos encontrar o bloco correspondente da mensagem original. Sem isto, não temos nenhum código útil. Entretanto, não é óbvio que isto é sempre verdade; faremos uma verificação cuidadosa na próxima seção.

Finalmente, desde o começo das notas vimos insistindo que codificamos usando n e decodificamos usando p e q; por isso é necessário fatorar n para

decodificar. Chegados a este ponto, vemos que isto não é estritamente verdadeiro. Além do próprio n, só precisamos conhecer o inverso d de e módulo $\phi(n)$ para decodificar. Acontece que só sabemos calcular d aplicando o algoritmo estendido a e e $\phi(n)$. Voltaremos a esta questão na seção 3.

No exemplo que estamos acompanhando, temos que $n = 143$ e $e = 7$. Aplicaremos o algoritmo euclidiano estendido para calcular d. Os cálculos são muito fáceis porque, dividindo $\phi(143) = 120$ por 7 obtemos

$$120 = 7 \cdot 17 + 1, \quad \text{donde} \quad 1 = 120 + (-17) \cdot 7.$$

Logo o inverso de 7 módulo 120 é -17. Como vamos usar d como expoente de potências, precisamos que d seja positivo. Portanto $d = 120 - 17 = 103$, que é o menor inteiro positivo congruente a -17 módulo 120. Assim, para decodificar o bloco 119 da mensagem codificada, calculamos a forma reduzida de 119^{103} módulo 143. Neste caso o cálculo já estará fora do nosso alcance se estamos limitados ao uso de lápis e papel. Usando um sistema de computação algébrica, podemos verificar que, de fato, $119^{103} \equiv 102 \pmod{143}$.

3. Por que funciona?

Como já observamos, o método exposto acima só será útil se, decodificando um bloco codificado, obtemos de volta o bloco correspondente da mensagem original. Digamos que temos um sistema RSA de parâmetros p e q, com $n = pq$. Então os dados de codificação serão n e e, e os de decodificação serão n e d. Na notação da seção 2, precisamos verificar que se b é um inteiro e $1 \leqslant b \leqslant n - 1$ então $\mathbf{DC}(b) = b$. Na verdade, vamos provar apenas que $\mathbf{DC}(b) \equiv b \pmod{n}$. Isto é suficiente porque tanto $\mathbf{DC}(b)$ quanto b estão no intervalo que vai de 1 a $n - 1$, logo só podem ser congruentes módulo n se são iguais. É por isso que precisamos escolher b menor que n, e é, também, por isso que temos que manter os blocos separados, mesmo depois da codificação.

Vamos aos cálculos. Por definição de $\mathbf{D}$ e $\mathbf{C}$ temos que

$$\mathbf{DC}(b) \equiv (b^e)^d \equiv b^{ed} \pmod{n}. \tag{3.1}$$

Porém d é o inverso de e módulo $\phi(n)$, logo $ed = 1 + k\phi(n)$, para algum inteiro k. Observe que, como e e d são inteiros maiores que 2 e $\phi(n) > 0$, então $k > 0$. Substituindo em (3.1)

$$b^{ed} \equiv b^{1+k\phi(n)} \equiv (b^{\phi(n)})^k b \pmod{n}.$$

Aqui o teorema de Euler vem em nosso socorro. Como $b^{\phi(n)} \equiv 1 \pmod{n}$,

resta-nos apenas que $b^{ed} \equiv b \pmod{n}$. Portanto

$$DC(b) \equiv b \pmod{n},$$

e a demonstração estaria completa, se não estivesse *'um pouco errada'*.

Se você leu o argumento acima com atenção, terá observado que apresenta um pequeno defeito. Se não notou nada errado, releia todo o parágrafo com cuidado, *levando em conta as hipóteses dos teoremas utilizados*.

Pois é, só um teorema é usado, o teorema de Euler. O problema é que só podemos usar este teorema para concluir que $b^{\phi(n)} \equiv 1 \pmod{n}$ se sabemos que $\text{mdc}(b, n) = 1$. Isto nem sempre é verdade, porque é muito difícil controlar quem são os números que são blocos da mensagem. Essencialmente qualquer número no intervalo 1 a $n - 1$ pode ocorrer. A solução é dar uma prova da congruência sem usar o teorema de Euler.

Lembre-se que $n = pq$, onde p e q são primos *distintos*. Calcularemos a forma reduzida de b^{ed} módulo p e módulo q. O cálculo é análogo para ambos os primos, logo basta executar um deles. Queremos, portanto, achar a forma reduzida de b^{ed} módulo p. Já vimos que

$$ed = 1 + k\phi(n) = 1 + k(p-1)(q-1),$$

logo

$$b^{ed} \equiv b \cdot (b^{p-1})^{k(q-1)} \pmod{p}.$$

Queremos usar o teorema de Fermat, mas para isto precisamos supor que p não divide b. Digamos que isto é verdade, então $b^{p-1} \equiv 1 \pmod{p}$ por Fermat, e obtemos $b^{ed} \equiv b \pmod{p}$.

Nada parece ter melhorado. Usamos Fermat, em vez de Euler, mas acabamos com o mesmo problema. Felizmente o fato de p ser primo nos permite tratar facilmente o caso em que p divide b. De fato, neste caso $b \equiv 0 \pmod{p}$ e a congruência é imediatamente verificada. Assim $b^{ed} \equiv b \pmod{p}$ vale para *qualquer valor de* b. Observe que *não* podemos usar um argumento semelhante diretamente para n porque $\text{mdc}(n, b) \neq 1$ não significa necessariamente que $b \equiv 0 \pmod{n}$, já que n é composto.

No parágrafo anterior vimos que $b^{ed} \equiv b \pmod{p}$ vale qualquer que seja p. Analogamente, podemos mostrar que $b^{ed} \equiv b \pmod{q}$. Em outras palavras, $b^{ed} - b$ é divisível por p e por q. Como p e q são primos distintos, temos que $\text{mdc}(p, q) = 1$, donde pq divide $b^{ed} - b$, pelo lema da seção 6 do capítulo 2. Como $n = pq$, concluímos que $b^{ed} \equiv b \pmod{n}$, para qualquer número inteiro b. Isto encerra a demonstração de que o método funciona. Só nos falta considerar a sua segurança.

4. Por que o RSA é seguro?

Lembre-se que o RSA é um método de chave pública. Sejam p e q os parâmetros do sistema que estamos usando e $n = pq$. A chave de codificação corresponde à chave pública do sistema. Portanto o par (n, e) é acessível a qualquer usuário. O RSA só será seguro se for difícil calcular d quando apenas n e e são conhecidos.

Na prática, só sabemos calcular d aplicando o algoritmo euclidiano estendido a $\phi(n)$ e e. Por outro lado, só sabemos calcular $\phi(n)$ se soubermos fatorar n para obter p e q. Portanto, na prática, só podemos quebrar o código se conseguirmos fatorar n. Mas sabemos que, se n for grande, este é um problema muito difícil, já que não são conhecidos algoritmos rápidos de fatoração.

Entretanto, nada nos impede de imaginar que alguém tenha inventado uma maneira de chegar a d sem ter que fatorar n. Por exemplo, o que aconteceria se alguém inventasse um algoritmo rápido para calcular $\phi(n)$ a partir de n e e ? Neste caso, teríamos na verdade obtido um algoritmo rápido de fatoração; vejamos porquê. O que estamos supondo é que $n = pq$ e $\phi(n) = (p-1)(q-1)$ são ambos conhecidos. Queremos determinar p e q a partir disto. Porém

$$\phi(n) = (p-1)(q-1) = pq - (p+q) + 1 = n - (p+q) + 1,$$

de forma que $p + q = n - \phi(n) + 1$ é conhecido. Contudo

$$(p+q)^2 - 4n = (p^2 + q^2 + 2pq) - 4pq = (p-q)^2;$$

logo

$$p - q = \sqrt{(p+q)^2 - 4n}$$

também é conhecido. Mas conhecendo $p + q$ e $p - q$ calculamos facilmente p e q; em outras palavras, fatoramos n.

O que vimos é que não adianta imaginar que alguém inventou uma maneira de achar $\phi(n)$ sem fatorar n, porque conhecendo os dois chegamos aos fatores de n. E se alguém inventar um algoritmo que ache d diretamente a partir de n e e. Como $ed \equiv 1 \pmod{\phi(n)}$, isto implica que conhecemos um múltiplo de $\phi(n)$. Isto também é suficiente para fatorar n, mas a demonstração está fora do alcance de um livro elementar como este. Para mais detalhes veja [**Rivest, Shamir e Adleman 1978,** §**IX.C**].

Só nos resta imaginar que seja possível achar b a partir da forma reduzida de b^e módulo n sem tentar achar d. Além de um processo de sim-

ples tentativa–impraticável quando n é grande–ninguém até agora conseguiu imaginar um tal método. Diante disto, acredita-se que quebrar o RSA e fatorar n sejam problemas equivalentes, embora até agora isto não tenha sido demonstrado.

5. Escolhendo primos

A questão da segurança do RSA é um pouco mais sutil do que a discussão da seção anterior deixa entrever. Um ponto muito importante refere-se à escolha dos primos p e q. É claro que se forem pequenos, o sistema será fácil de quebrar. Mas não basta escolhê-los grandes. De fato, se p e q são grandes, mas $|p - q|$ é pequeno, então é fácil fatorar $n = pq$ usando o algoritmo de Fermat, apresentado na seção 4 do capítulo 2.

Isto não é puro papo furado. Em 1995 dois estudantes de uma universidade americana quebraram uma versão do RSA em uso público. O feito só foi possível porque a escolha dos primos usada neste sistema era inteiramente inadequada. Por outro lado, o RSA está em uso há anos e, se os primos forem bem escolhidos, o sistema tem-se mostrado muito seguro. Portanto, uma receita para escolher primos bons é essencial para a "caixa de ferramentas" de qualquer um que deseje programar o RSA.

Suponha que desejamos implementar o RSA com chave pública (n, e), de modo que n seja um inteiro com aproximadamente r algarismos. Para construir n, escolha um primo p entre $4r/10$ e $45r/100$ algarismos e, em seguida, escolha q próximo de $10^r/p$. O tamanho de chave atualmente recomendado para uso pessoal é 768 bits. Isto significa que n terá aproximadamente 231 algarismos. Para construir um tal n precisaremos de dois primos de, digamos 104 e 127 algarismos, respectivamente. Note que estes primos estão longe bastante um do outro para fazer com que a fatoração de n pelo algoritmo de Fermat seja impraticável. Contudo, precisamos também ter certeza que os números $p - 1$, $q - 1$, $p + 1$ e $q + 1$ não têm fatores primos pequenos, porque isto faria de n uma presa fácil para alguns dos algoritmos de fatoração conhecidos; mais detalhes podem ser encontrados em [**Riesel 1994, capítulo 6, pp. 174-7**]. Vejamos, então, como primos tão grandes podem ser obtidos.

Em primeiro lugar precisamos de um resultado simples sobre a distribuição dos primos. Lembre-se que $\pi(x)$ denota o número de primos positivos menores ou iguais a x. De acordo com o teorema dos números primos, se x é bastante grande, então $\pi(x)$ é aproximadamente igual a $x/\log x$, onde $\log$ é o logaritmo em base e; veja seção 5 do capítulo 3. Seja agora x um número muito grande e ϵ um número positivo qualquer. Queremos obter

uma aproximação para o número de primos entre x e $x + \epsilon$; isto é, para $\pi(x + \epsilon) - \pi(x)$. Segue do teorema dos números primos, e das propriedades do logaritmo, que $\pi(x + \epsilon) - \pi(x)$ é aproximadamente igual a

$$\frac{x + \epsilon}{\log x + \log(1 + x^{-1}\epsilon)} - \frac{x}{\log x}$$

Supondo que $x^{-1}\epsilon$ é pequeno, podemos substituir $\log(1 + x^{-1}\epsilon)$ por zero e ainda assim obter uma aproximação razoável para $\pi(x + \epsilon) - \pi(x)$. Concluímos que o número de primos entre x e $x + \epsilon$ é aproximadamete igual a $\epsilon/\log x$. É claro que, quanto maior for x, menor será $x^{-1}\epsilon$, e melhor será a aproximação. Para uma discussão mais detalhada desta aproximação consulte [**Hardy and Wright 1994, capítulo XXII, seção 22.19**].

Suponha, agora, que desejamos achar um primo próximo de um inteiro positivo x. Para tornar a discussão mais concreta, digamos que x é da ordem de 10^{127}. Procuraremos por este primo no intervalo que vai de x a $x + 10^4$. Seria útil saber com antecedência quantos primos devemos esperar que haja neste intervalo. É aqui que o resultado do parágrafo anterior vem em nosso auxílio. Note que, neste exemplo, $x^{-1}\epsilon$ é da ordem de 10^{-123} e, portanto, é bem pequeno. Assim, usando a fórmula acima, concluímos que no intervalo que vai de x a $x + 10^4$ deve haver aproximadamente

$$[10^4/\log(10^{127})] = 34$$

primos. No final da seção 2 do capítulo 10, esboçamos uma estratégia para verificar se um dado inteiro ímpar n é primo. Esta estratégia consiste de três etapas:

(1) Verifique se n é divisível por um primo menor que 5000.
(2) Supondo que n não é divisível por nenhum destes primos, aplique o teste de Miller a n usando como bases os primeiros 10 primos.
(3) Supondo que o teste de Miller teve como saída inconclusivo para todas estas bases, aplique o teste de primalidade da seção 2, do capítulo 10, a n.

Adaptaremos esta estratégia para achar um primo no intervalo que vai de x a $x + 10^4$. Em primeiro lugar, elimine do intervalo dado os inteiros ímpares que são divisíveis por primos menores que $5 \cdot 10^3$. Em seguida, aplique (2) e (3) aos números remanescentes, até que um primo seja encontrado.

Para descobrir quanto trabalho teremos para obter este primo, podemos tentar determinar aproximadamente quantos inteiros sobraram depois que eliminamos aqueles que são divisíveis por primos menores que $5 \cdot 10^3$. Seja

m um inteiro positivo. Se $x \leqslant km \leqslant x + 10^4$, então

$$[\frac{x}{m}] \leqslant k \leqslant [\frac{x + 10^4}{m}].$$

Assim, há

$$[\frac{x + 10^4}{m}] - [\frac{x}{m}]$$

múltiplos de m no intervalo que vai de x a $x + 10^4$. Este número é aproximadamente igual a $[10^4/m]$, que é o número de múltiplos de m menores que 10^4. Isto implica que o número de inteiros que são múltiplos de primos positivos menores que $5 \cdot 10^3$ nos intervalos $[x, x + 10^4]$ e $[0, 10^4]$ é aproximadamente o mesmo. Não é difícil calcular o último destes números. Em primeiro lugar, note que um número composto menor que 10^4 tem que ser múltiplo de um primo menor que $\sqrt{10^4} = 100$. Assim, um inteiro no intervalo $[0, 10^4]$ é múltiplo de um primo menor que $5 \cdot 10^3$ se é composto, ou se é ele próprio um primo menor que $5 \cdot 10^3$. Levando em conta que 2 é o único primo par, temos que o número de inteiros *compostos e ímpares* menores que 10^4 é $5000 - \pi(10^4) + 1$. Dessa forma, o número total de inteiros ímpares que restam no intervalo de x a $x + 10^4$ após a eliminação daqueles que são divisíveis por primos menores que $5 \cdot 10^3$ é aproximadamente igual a

$$5000 - (5000 - (\pi(10^4) - 1)) - (\pi(5 \cdot 10^3) - 1) = 560.$$

Observe que $\pi(10^4)$ e $\pi(5 \cdot 10^3)$ são facilmente calculados usando o crivo de Eratóstenes.

Assim, esperaríamos encontrar, em média 34 primos de um total de 560 inteiros deixados após o processo inicial de eliminação.

6. Assinaturas

Imagine que uma empresa realiza suas transações bancárias por computador, via rede telefônica. É evidente que, por razões de segurança, tanto a empresa quanto o banco vão exigir que a informação seja codificada, antes de ser enviada pelo sistema telefônico. Mas isto só não basta. Afinal se o RSA está sendo usado, então os dados de codificação do banco são públicos. Assim, em princípio, qualquer um pode mandar uma mensagem codificada ao banco relativa ao movimento financeiro da empresa. Por exemplo, um *'hacker'* poderia facilmente mandar instruções ao banco para que o saldo bancário da empresa fosse transferido para uma outra conta. Por isso o banco precisa de uma garantia de que a mensagem teve origem em um usuário autorizado

da empresa. Ou seja, a mensagem tem que ser *'assinada'*: uma assinatura eletrônica, é claro.

Não é difícil mandar uma mensagem assinada usando o RSA, ou qualquer outro sistema de chave pública. Vamos chamar de C_e e D_e as funções codificação e decodificação da empresa; e de C_b e D_b as funções correspondentes do banco. Seja a um bloco da mensagem que a empresa quer enviar ao banco. De acordo com o que vimos antes, a empresa deveria codificar a como $C_b(a)$ e enviá-lo por linha telefônica. Para mandar a mensagem assinada, ao invés de $C_b(a)$ enviamos $C_b(D_e(a))$. Isto é, primeiro aplicamos a função decoficação da empresa a a e só depois codificamos o bloco usando a função codificação do banco.

Tendo recebido o bloco $C_b(D_e(a))$, o banco aplica sua função de decodificação para obter $D_e(a)$, e a este último bloco aplica a função codificação da empresa para obter a, que é o bloco original. Observe que C_e é público, por isso é conhecido do banco.

Por que isto é suficiente para garantir ao banco que a mensagem tem origem autorizada? O banco aplica aos blocos da mensagem recebida a seqüência de funções C_eD_b. Se a mensagem resultante fizer sentido, isto significa que a mensagem enviada foi codificada aplicando aos blocos da mensagem original as funções C_bD_e. Mas lembre-se que D_e só é conhecido da empresa. Portanto, se a mensagem fizer sentido, tem que ter tido origem na empresa! A probabilidade de que uma mensagem produzida sem usar D_e seja decodificada pelo banco (usando C_eD_b) e produza uma mensagem com sentido é praticamente zero. Assim o banco pode estar seguro de que a mensagem é legítima.

Recentemente descobriu-se que o uso pouco cuidadoso da técnica de autenticação de assinaturas torna vulneráveis certos métodos de chave pública como o RSA. No final de 1995, um consultor em assuntos de segurança de computadores (mas formado em biologia!) descobriu que é possível usar o sistema de assinaturas para quebrar o RSA. O método consiste em enviar uma mensagem assinada e marcar o tempo que o sistema leva para confirmar a assinatura. Fazendo isto para mensagens de tamanhos ligeiramente diferentes, é possível obter informações suficientes para encontrar a chave de decodificação do sistema RSA que esteja sendo usado. Portanto, a segurança do RSA não depende exclusivamente da nossa capacidade de inventar novos algoritmos de fatoração. Há muitos outros fatores importantes, que não têm um caráter puramente matemático.

7. Exercícios

1. Sabendo-se que $n = 3552377$ é igual ao produto de dois números primos e que $\phi(n) = 3548580$; fatore n.

2. A chave pública utilizada pelo Banco de Toulouse para codificar suas mensagens é a seguinte: $n = 10403$ e $e = 8743$. Recentemente os computadores do banco receberam, de local indeterminado, a seguinte mensagem:

$$4746 - 8214 - 9372 - 9009 - 4453 - 8198$$

O que diz a mensagem mandada ao Banco de Toulouse ?

3. A mensagem $6355 - 5075$ foi codificada pelo método RSA usando a senha $n = 7597$ e $e = 4947$. Além disso, sabe-se que $\phi(n) = 7420$. decodifique a mensagem.

4. Sejam p e q primos ímpares e digamos que estamos trabalhando com uma implementação do RSA com chave de codificação (n, e), onde $n = pq$. Pode acontecer que um bloco b de uma mensagem seja codificado como ele próprio nesta implementação. Ou seja, pode acontecer que $C(b) = b$. Um tal bloco será chamado de *invariante* pelo RSA com chave (n, e). Determine quantos são os blocos invariantes pelo RSA quando $p = 3$, $q > 3$ e $e = 3$.

Sugestão: Se b é invariante pelo RSA com chave (n, e) então $b^e \equiv b \pmod{n}$. Portanto $b^e \equiv b \pmod{p}$ e $b^e \equiv b \pmod{q}$. Resolva cada uma destas equações quando $p = 3$, $q > 3$ e $e = 3$ e 'cole' os resultados usando o teorema chinês do resto.

5. Um outro método de criptografia de chave pública bem conhecido é o *El Gamal*. Para construir este sistema é necessário escolher um primo p muito grande, e um gerador $\overline{g}$ de $U(p)$. Estes dados serão comuns a todos os usuários de uma dada implementação do sistema. Cada usuário escolhe, então, um inteiro entre 0 e $p - 1$. Este número deverá ser mantido secreto, já que será a chave de decodificação daquele usuário. Se um usuário escolheu como chave de decodificação o número a, então sua chave de codificação (pública) será $\overline{g}^a$. Seja b um bloco da mensagem a ser enviada; devemos ter que $1 \leqslant b \leqslant p - 1$. Para codificar o bloco b, escolha aleatoriamente um inteiro positivo k e envie o par $(\overline{g}^k, \overline{b}\overline{g}^{ak})$.
(1) Por que é fácil decodificar a mensagem quando conhecemos a e $\overline{g}$?
(2) O que é necessário fazer para 'quebrar' este código?

A resposta a (2) é conhecida como *problema do logaritmo discreto*. Acredita-se que este problema seja tão difícil quanto o de fatorar um inteiro grande.

6. O método de criptografia inventado por Michael Rabin em 1979 é muito semelhante ao RSA. Para começar, escolha primos ímpares distintos p e q, e ponha $n = pq$. Como no RSA, há uma etapa de pré-codificação, em que a mensagem é convertida em um número, que é então dividido em blocos menores que n. Vamos nos limitar a explicar como codificar e decodificar um bloco b, onde $0 \leqslant b < n$. Para *codificar* b calcule o resto da divisão de b^2 por n. Por outro lado, se a é um bloco da mensagem codificada, então para *decodificar* a basta resolver a equação $x^2 \equiv a \pmod{n}$. Seja

$$u = (p^{q-1} - q^{p-1}).$$

(1) Mostre que $u^2 \equiv 1 \pmod{q}$ e $u^2 \equiv 1 \pmod{p}$.
(2) Mostre que $u^2 \equiv 1 \pmod{n}$.
(3) Mostre que, se x_0 é uma solução de $x^2 \equiv a \pmod{n}$, então $-x_0$, ux_0 e $-ux_0$ também são soluções.

Isto significa que cada bloco admite 4 decodificações possíveis! Isto é uma desvantagem em relação ao RSA, que só produz uma decodificação para cada bloco.

7. Seja $n = pq$ onde p e q são primos ímpares distintos. Vimos na seção 4 que decifrar o RSA é provavelmente equivalente a fatorar n. Vamos demonstrar que isto é verdade para o método de Rabin, descrito no exercício 6. Em outras palavras, queremos mostrar que se dispomos de uma máquina que, dado a, determina uma solução de $x^2 \equiv a \pmod{n}$, então podemos facilmente fatorar n e achar p e q. Para fazer isto, escolha b *aleatoriamente* e calcule a forma reduzida de b^2 módulo n. Use então a máquina para achar uma solução de $x^2 \equiv b^2 \pmod{n}$. Como há 4 soluções possíveis, temos uma probabilidade de $1/2$ de que a máquina nos fornecerá uma solução x tal que $x \not\equiv \pm b \pmod{n}$. Mostre que, se for este o caso, então $\mathrm{mdc}(x, b)$ tem que ser p ou q. Isto significa que, se dispomos de uma máquina como esta, então esperaríamos encontrar p e q efetuando apenas poucas escolhas para b.

8. Sejam p e q primos que deixam resto 3 na divisão por 4, e ponha $n = pq$. Digamos que a é um bloco de uma mensagem codificada pelo método de Rabin usando n como parâmetro. Escreva um programa que, tendo por entrada n e a, acha todas as possíveis decodificações de a. Para isto vai ser preciso:

(1) fatorar n para achar p e q, e
(2) resolver a equação $x^2 \equiv a \pmod{n}$.

As duas etapas já foram objeto de exercícios anteriores; a saber, exercícios 12 do capítulo 2, 17 do capítulo 5 e 10 do capítulo 7. Use o seu programa para decodificar a mensagem dada a seguir, sabendo-se que o método de Rabin utilizado tem parâmetro $n = 20490901$:

$$18504501 - 18109884.$$

Você vai obter $4 \cdot 4 = 16$ decodificações possíveis, mas só uma vai fazer sentido.

Epílogo

Chegados ao fim do livro nos vemos forçados a reconhecer que apenas arranhamos a superfície de um enorme monolito. Ou, de um ponto de vista mais positivo, apenas começamos a assentar as fundações sobre as quais se apóiam várias áreas da matemática. Separando as linhas dos vários temas que desenvolvemos, é possível reconhecer claramente os seguintes tópicos:

- métodos matemáticos da criptografia;
- teoria algorítmica de números;
- teoria de grupos;
- computação algébrica.

1. Criptografia e teoria de números

É muito difícil separar os dois primeiros tópicos; por isso vamos tratá-los em conjunto. Embora tenhamos estudado com um certo cuidado vários métodos para determinar se um número é primo (ou composto), nosso tratamento dos algoritmos de fatoração foi extremamente incompleto. Além do algoritmo que procede por tentativa sistemática, consideramos apenas o algoritmo de Fermat, que já tem uns 300 anos de idade. Entre os algoritmos mais modernos, dois claramente se destacam: o algoritmo da *curva elítica* e o *crivo quadrático*. O crivo quadrático só se torna eficiente para inteiros com mais de 25 algarismos. Aos inteiros grandes demais para fatorar por tentativa, mas pequenos demais para o crivo quadrático, aplica-se o algoritmo da curva elítica.

Muitas fatorações recentes de números espetacularmente grandes foram obtidas usando o crivo quadrático. Por exemplo, pouco depois do RSA ser inventado, uma mensagem desafio foi codificada usando uma chave pública de 129 algarismos; e que, por isso, ficou conhecida como RSA-129. Acreditava-se que, com os algoritmos e *hardware* disponíveis na década de 70, não seria possível decifrar esta mensagem em menos de alguns quadrilhões de anos. Além disso, não se esperava que fosse possível vir a quebrar esta mensagem antes do final do século. Entretanto, usando o crivo quadrático, 600 computadores de voluntários espalhados por 25 países e um supercomputador para

reunir tudo, foi possível fatorar a chave e quebrar o código em 1994. O trabalho levou oito meses para ser concluído; o conteúdo da mensagem era:*'the magic words are squeamish ossifrage'*.

Estes dois algoritmos estão longe de esgotar o arsenal dos criptoanalistas. Há também o algoritmo das *frações continuadas* e o *crivo do corpos de números*. Todos estes algoritmos dependem de métodos matemáticos que não foram estudados neste livro. Para um tratamento dos algoritmos da curva elítica, das frações continuadas e do crivo quadrático, juntamente com seus pré-requisitos matemáticos, consulte [**Bressoud 1989**]. Para uma bela introdução à teoria das curvas elíticas, que inclui o algoritmo de fatoração, consulte [**Silverman e Tate 1992**].

O crivo dos corpos de números, que foi usado na fatoração do número de Fermat $F(9)$ em 1993, é discutido em [**Riesel 1994**] e em [**Lenstra, Lenstra, Manasse e Pollard 1993**]. Este algoritmo foi usado recentemente para quebrar o sucessor do RSA-129, conhecido como RSA-130, porque a chave pública tem 130 algarismos. É tal a eficiência deste crivo, que o tempo necessário para fatorar a chave pública e, portanto, decifrar o RSA-130, ficou em torno de 15% do tempo utilizado para quebrar o código anterior. Para mais detalhes, veja [**Pomerance 1996**].

2. Teoria de grupos

O terceiro dos tópicos mencionados acima é, na verdade, uma das áreas mais ricas e ativas da matemática. Como vimos no capítulo 8, a teoria de grupos nasceu intimamente ligada à teoria das equações polinomiais. Esta faceta da teoria culminou com o critério obtido por Galois, que permite determinar se uma equação admite ou não uma fórmula algébrica, a partir de um grupo finito a ela associado. Trata-se da *teoria de Galois*, uma das áreas mais elegantes da álgebra. Para mais detalhes a este respeito, consulte [**Gonçalves 1979**] ou [**Artin 1991, capítulo 14**]. Uma história da teoria de equações, que inclui uma exposição completa da teoria de Galois, pode ser encontrada em [**Edwards 1984**].

O estudo das simetrias é uma outra aplicação importante da teoria de grupos. Por exemplo, usando grupos podemos mostrar que os únicos poliedros regulares são os cinco sólidos platônicos: o tetraedro, o cubo, o octaedro, o dodecaedro e o iscosaedro. Podemos também classificar todas as possíveis maneiras de ladrilhar o plano; o que, por sua vez, conduz a uma série de aplicações à cristalografia. Para os detalhes, consulte [**Artin 1991, capítulo 5**].

Uma visão de conjunto das relações entre grupos e simetria, narrada de maneira magistral por um dos maiores matemáticos deste século, encontra-se no livro *Symmetry* de H. Weyl, [**Weyl 1982**]. Neste livro você encontrará o comentário sobre a obra de Galois mencionado na seção 3 do capítulo 8.

Finalmente a teoria de grupos é, por si, uma área extremamente atraente da matemática. Os grupos combinam uma grande economia na definição, com uma enorme variedade de exemplos. Pode-se provar, por exemplo, que os grupos finitos admitem uma espécie de 'teorema de fatoração', com o papel dos números primos sendo desempenhado pelos chamados *grupos simples*. Em 1980 foi concluído um programa de vários anos que levou à catalogação de todos os grupos simples finitos. A demonstração deste teorema se estende por milhares de páginas ao longo de mais de dez anos. Muitos grupos simples se agrupam em famílias (três famílias, ao todo); além destes há apenas os 26 grupos esporádicos. O maior destes é conhecido como *o monstro*, porque sua ordem é aproximadamente 10^{53}. Não é surpreendente que os computadores tenham desempenhado um papel importante na construção de grupos de ordem tão grande. Para uma amostra de como o computador pode ser usado na teoria de grupos consulte [**Neubuser e Sidki 1988**].

3. Computação algébrica

O último dos tópicos relacionados acima é computação algébrica. Na verdade, muito do que já dissemos faz parte do arsenal de técnicas da computação algébrica. Mas há um tópico especial que ficou de fora e que não podemos esquecer: as técnicas computacionais usadas na teoria de polinômios. Duas facetas precisam ser consideradas. A teoria dos polinômios em uma variável, com coeficientes em $\mathbf{Q}$ digamos, é muito semelhante à teoria de números. Isto é conseqüência da existência de um algoritmo de divisão para estes polinômios. Podemos usar o algoritmo de divisão, exatamente como fizemos nos capítulos 1 e 2, para provar um teorema de fatoração única. O papel de primos é desempenhado pelos polinômios irredutíveis. A surpresa é que, ao contrário do que acontece com os inteiros, *existe* um algoritmo eficiente para fatorar polinômios! Para uma introdução a este tipo de problema consulte [**Knuth 1981**] e [**Davenport, Siret e Tournier 1988**].

Em contrapartida não existe um algoritmo de divisão para os polinômios em *várias indeterminadas* com coeficientes em $\mathbf{Q}$. Mesmo assim estes polinômios admitem um teorema de fatoração única. A demonstração, naturalmente, é bem mais complicada e nenhum algoritmo eficiente para efetuar a fatoração é conhecido. Entretanto os polinômios em várias indetermina-

das são muito importantes em geometria; o que levou ao desenvolvimento de vários algoritmos muito úteis. Estes algoritmos podem ser aplicados, por exemplo, na demonstração por computador de teoremas de geometria, na teoria de grafos (coloração de mapas) e em questões de robótica. Para maiores detalhes consulte [**Costa e Vainsencher 1995**] e [**Cox, Little e O'Shea 1992**].

Finalmente um comentário dirigido aos entusiastas, e aos aventureiros que sonhem embarcar em uma jornada de descobrimento. Uma nova era pode estar se abrindo na criptografia, no fantástico casamento entre teoria de números, computação e mecânica quântica. De uns anos para cá alguns físicos observaram que é possível usar propriedades quânticas da matéria na construção de computadores que teriam várias características diferentes dos atuais. A principal novidade estaria na utilização sistemática do chamado *princípio de superposição*; que é discutido, por exemplo, em [**Feynman 1985**]. Durante muito tempo isto foi visto como pouco mais do que um exercício de física teórica, já que os algoritmos tipicamente 'quânticos' resolviam apenas problemas bastante artificiais. A situação mudou em 1994, quando Peter Shor (do AT&T Bell Labs) mostrou que se um computador quântico puder ser construído então teremos uma maneira de fatorar inteiros enormes de maneira extremamente rápida; veja [**Shor 1994**]. Isto tornaria o RSA totalmente obsoleto! Trabalhos recentes como o de Shor levaram a uma nova onda de idéias de como resolver os problemas tecnológicos necessários para fazer funcionar um computador quântico. Se estas idéias darão frutos ainda é cedo para saber. Da História, porém, podemos tirar sem receio a conclusão de que é de desenvolvimentos impredizíveis como este que será construído o futuro.

Apêndice

Algoritmos Complementares

Neste apêndice apresentamos dois algoritmos que são necessários na implementação do algoritmo de fatoração de Fermat e de muitos dos testes de primalidade que estudamos neste livro. Um dos algoritmos calcula a parte inteira da raiz quadrada de um inteiro positivo dado; o outro calcula potências em aritmética modular.

1. Raízes quadradas

No capítulo 2, seções 2 e 3 estudamos dois algoritmos de fatoração. Para aplicá-los é necessário saber calcular a raiz quadrada de certos inteiros positivos. Isto é fácil de fazer com uma calculadora, se os números são pequenos. Mas como proceder se desejamos implementar o algoritmo? Nesta seção descrevemos um método para calcular a parte inteira da raiz quadrada de um inteiro positivo dado. Isto é suficiente para os propósitos de ambos os algoritmos. Por exemplo, para mostrar, no algoritmo de Fermat, que um inteiro positivo n é um quadrado perfeito basta verificar que $n - [\sqrt{n}]^2 = 0$.

O método que descreveremos consiste em calcular uma seqüência de inteiros positivos

$$x_0, x_1, x_2, \ldots$$

tal que, se $x_k \leqslant x_{k+1}$ para algum $k \geqslant 0$, então $x_k = [\sqrt{n}]$. A seqüência é definida recursivamente por

$$x_0 = \left[\frac{n+1}{2}\right] \quad \text{e} \quad x_{i+1} = \left[\frac{x_i^2 + n}{2x_i}\right],$$

para $i \geqslant 0$. A idéia é que, começando em $(n+1)/2$, a seqüência decrescerá até atingir $[\sqrt{n}]$. Considere, agora, as seguintes afirmações:

(1) $[(n+1)/2] \geqslant [\sqrt{n}]$,

(2) se $x_k > [\sqrt{n}]$, então $x_k > x_{k+1}$, e

(3) se $x_k \leqslant x_{k+1}$, então $x_k = [\sqrt{n}]$.

Assim, cada x_k é um inteiro positivo menor ou igual a $(n+1)/2$. Portanto, só pode haver uma quantidade finita de elementos distintos na seqüência. Em particular, tem que existir algum r para o qual $x_r \leqslant x_{r+1}$, e (3) então implica que $x_r = [\sqrt{n}]$.

Com isso, precisamos apenas provar as três afirmações acima. Note que, se $y \geqslant x$ são números reais, então $[y] \geqslant [x]$. Assim, (1) segue se provarmos que $(n+1)/2 > \sqrt{n}$. Mas esta última desigualdade é uma conseqüência imediata de

$$\frac{(n+1)^2}{2} = \frac{n^2+2n+1}{2} > n^2.$$

Para provar (2) lembre-se que, por definição $x_{k+1} = [(x_k^2+n)/2x_k]$. Como x_k é um inteiro, segue de $x_k > [\sqrt{n}]$, que $x_k > \sqrt{n}$. Elevando ambos os membros ao quadrado, obtemos $x_k^2 > n$. Portanto, $2x_k^2 > x_k^2 + n$, de modo que

$$x_k > \frac{x_k^2+n}{2x_k} \geqslant x_{k+1}.$$

Para provar (3), suponha que $x_k \leqslant [(x_k^2+n)/2x_k]$. Isto é equivalente a

$$0 \leqslant \frac{x_k^2+n}{2x_k} - x_k < 1,$$

que implica

$$0 \leqslant n - x_k^2 < 2x_k.$$

Portanto, $n \geqslant x_k^2$ e $n < x_k^2 + 2x_k$. Mas, desta última desigualdade obtemos $n < (x_k+1)^2$. Assim,

$$x_k^2 \leqslant n < (x_k+1)^2,$$

que implica que $x_k = [\sqrt{n}]$, provando (3).

Este método de achar a parte inteira de uma raiz quadrada pode ser facilmente implementado, como mostra o seguinte algoritmo.

Algoritmo da raiz quadrada.
Entrada: *um inteiro* $n > 2$.
Saída: *a parte inteira da raiz quadrada de* n.

Etapa 1: *Comece fazendo* $X = (n+1)/2$ *e* $Y = [(n+1)/2]$, *e vá para a Etapa 2;*

Etapa 2: *Se* $X \leqslant Y$, *páre e escreva a parte inteira da raiz quadrada de* n *é igual a* X; *senão vá para a Etapa 3;*

Etapa 3: *Substitua o valor de* X *pelo de* Y*, e o de* Y *por* $[(X^2+n)/2X]$*, e volte à Etapa 2.*

2. Potenciação módulo n

Suponhamos que são dados três inteiros positivos a, e e n. O algoritmo que descreveremos nesta seção calcula a forma reduzida de a^e módulo n. Para isto usaremos a expansão binária do expoente, o que permite efetuar a potenciação de maneira muito rápida.

Digamos que a expansão binária do expoente é

$$e = b_n 2^n + \cdots + b_1 2 + b_0,$$

onde os coeficientes $b_0, b_1, \ldots, b_n$ valem 0 ou 1. Temos então que

$$a^e = (a^2)^{b_n 2^{n-1} + \cdots + b_2 2 + b_1} \cdot a^{b_0}.$$

Observe que a^{b_0} pode valer 1 (se $b_0 = 0$) ou a (se $b_0 = 1$). Seja $P_1 = a^{b_0}$; continuando, temos

$$a^e = (a^4)^{b_n 2^{n-2} + \cdots + b_3 2 + b_2} \cdot (a^2)^{b_1} P_1.$$

Seja $P_2 = (a^2)^{b_1} P_1$; temos então

$$a^e = (a^8)^{b_n 2^{n-3} + \cdots + b_4 2 + b_3} \cdot (a^4)^{b_2} P_2.$$

Continuando assim, teremos uma seqüência de inteiros $P_1, P_2, \ldots P_n$ onde $P_n = a^e$. Naturalmente se estamos calculando em $\mathbb{Z}_n$, efetuaremos a redução módulo n de cada uma das multiplicações que aparecer no algoritmo.

Observe que a cada etapa há apenas duas multiplicações básicas:

(1) o cálculo de um quadrado para obter a seqüência de potências;
(2) o produto $a^{2^i} P_i$, para $i = 1, 2, \ldots, n$.

Além disso, se na etapa i temos $b_i = 0$, então não é necessário efetuar o produto $a^{2^i} P_i$.

Na prática o algoritmo determina a expansão binária de e à medida que calcula a potência. De fato, se e é ímpar então $b_0 = 1$; se par, então $b_0 = 0$. Podemos determinar b_1 de maneira semelhante, analisando a paridade de

$$b_n 2^{n-1} + \cdots + b_2 2 + b_1.$$

Observe que este último número é $e/2$, se e é par, e $(e-1)/2$ se e é ímpar; e assim por diante. O algoritmo é o seguinte:

Algoritmo de exponenciação.
Entrada: *Inteiros* a, e *e* n, *onde* $a, n > 0$ *e* $e \geqslant 0$.
Saída: *a forma reduzida de* a^e *módulo* n.

Etapa 1: *Comece com* $A = a$, $P = 1$ *e* $E = e$;
Etapa 2: *Se* $E = 0$ *imprime* '$a^e \equiv P \pmod{n}$'; *senão vai para a Etapa 3;*
Etapa 3: *Se* E *for ímpar então atribua a* P *o valor do resto da divisão de* $A \cdot P$ *por* n *e a* E *o valor* $(E-1)/2$ *e vá para Etapa 5, senão vá para a Etapa 4;*
Etapa 4: *Se* E *for par então atribua a* E *o valor* $E/2$ *e vá para a Etapa 5;*
Etapa 5: *Substitua o valor atual de* A *pelo resto da divisão de* A^2 *por* n *e vá para a Etapa 2.*

Na prática, este algoritmo é muito eficiente no cálculo de potências módulo n; além de ser muito simples de implementar.

Bibliografia

Alford, W.R., Granville, A. and Pomerance, C. (**1994**) There are infinitely many Carmichael numbers, *Ann. of Math.* **140**, 703–722.

Arnault, F. (**1995**) Constructing Carmichael numbers which are strong pseudoprimes to several bases, *J. Symbolic Computation* **20**, 151–161.

Artin, M. (**1991**) *Algebra*, Prentice-Hall, Englewood Cliffs.

Boyer, C.B. (**1994**) *História da Matemática*, traduzido por E.F. Gomide, Editora Edgar-Blücher, São Paulo.

Bressoud, D.M. (**1989**) *Factorization and primality testing*, Undergraduate Texts in Mathematics, Springer-Verlag, New York.

Bruce, J.W. (**1993**) A really trivial proof of the Lucas-Lehmer test, *Amer. Math. Monthly* **100**, 370–371.

Carmichael, R. D. (**1912**) On composite numbers P which satisfy the Fermat congruence $a^{P-1} \equiv 1 \pmod{P}$, *Amer. Math. Monthly* **19**, 22–27.

Costa, A.V. e Vainsencher, I. (**1995**) Bases de Gröbner: resolvendo equações polinomiais, em *Atas da XIII Escola de Álgebra*, vol. I, IMECC-UniCamp, Campinas.

Cox, D., Little, J. e O'Shea, D. (**1992**) *Ideals, varieties and algorithms: an introduction to computation algebraic geometry and commutative algebra*, Undergraduate Texts in Mathematics, Springer-Verlag, New York.

Davenport, J.H., Siret, Y. e Tournier, E. (**1988**) *Computer Algebra: systems and algorithms for algebraic computation*, translated by A. Davenport and J.H. Davenport, Academic Press, London.

Davis, M. (**1980**) What is computation?, in *Mathematics Today*, editado por L. A. Steen, Vintage Books, New York, 241–267.

Dickson, L.E. (**1952**) *A history of the theory of numbers*, Chelsea Publishing Company, New York.

Edwards, H.M. (**1977**) *Fermat's last theorem*, Graduate Texts in Mathematics 50, Springer-Verlag, New York.

Edwards, H.M. (**1984**) *Galois theory*, Graduate Texts in Mathematics 101, Springer-Verlag, New York.

Feynman, R.P. (**1985**) *QED: the strange theory of light and matter*, Princeton University Press, Princeton.

Gauss, C.F. (**1986**) *Disquisitiones Arithmeticæ*, translated by A.A. Clarke, revised by W.C. Waterhouse with the help of C. Greither e A.W. Grotendorst, Springer-Verlag, New York.

Giblin, P. (**1993**) *Primes and programming*, Cambridge University Press, Cambridge.

Gonçalves, A. (**1979**) *Introdução à álgebra*, Projeto Euclides, IMPA, Rio de Janeiro.

Gostin, G.B. (**1995**) New factors of Fermat numbers, *Math. of Comp.* **64**, 393–395.

Hardy, G.H. e E.M. Wright (**1994**) *An introduction to the theory of numbers*, quinta edição, Oxford Science Publications, Oxford University Press, Oxford.

Hefez, A. (**1993**) *Curso de Álgebra*, vol. 1, Coleção Matemática Universitária, IMPA, Rio de Janeiro.

Ingham, A. E. (**1932**) *The distribution of primes*, Cambridge University Press, Cambridge.

Knuth, D.E. (**1981**) *The Art of computer programming*, vol. 2, *Seminumerical algorithms*, segunda edição, Addison-Wesley Publishing Company, Reading.

Koblitz, N. (**1987**) *A course in number theory and criptography*, Graduate Texts in Mathematics 97, Springer-Verlag, New York.

Kranakis, E. (**1986**) *Primality and criptography*, Wiley-Teubner series in Computer Science, B.G. Teubner e J. Wiley & Sons.

Lemos, M. (**1989**) *Criptografia, números primos e algoritmos*, 17° Colóquio Brasileiro de Matemática, IMPA/CNPq.

Lenstra, A.K., Lenstra jr, H.W., Manasse, M.S. e Pollard, J.M. (**1993**) The factorization of the ninth Fermat number, *Math. of Comp.* **61**, 319–349.

Neubuser, J. e Sidki, S. (**1988**) Alguns procedimentos computacionais para grupos dados por uma apresentação finita, *Matemática Universitária*, número **7**, 77–120.

Pomerance, C. (**1996**) A tale of two sieves, *Notices of the Amer. Math. Soc.* **43**(12), 1473–1485.

Ramanujan, S. (**1927**) *Collected papers of Srinivasa Ramanujan*, editado por G. H. Hardy, P. V. Seshu Aiyar e B. M. Wilson, Cambridge University Press, Cambridge.

Ribenboim, P. (**1990**) *The book of prime number records*, Springer-Verlag, New York.

Ribenboim, P. (**1996**) A conjectura de Catalan, *Matemática Universitária*, número **20/21**, 3–16.

Riesel, H. (**1994**) *Prime numbers and computer methods of factorization*, second edition, Progress in Mathematics 126, Birkhäuser, Boston.

Rivest, R.L., Shamir, A. e Adleman, L. (**1978**) A method for obtaining digital signatures and public-key cryptosystems, *Comm. ACM*, **21**, 120–126.

Shor, P.W. (**1994**) Algorithms for quantum computation: discrete logarithms and factoring, in *Proceedings of 35th Annual Symposium on Foundations of Computer Science*, IEEE Computer Science Press, 124–134.

Silverman, J.H. e Tate, J. (**1992**) *Rational points on elliptic curves*, Undergraduate Texts in Mathematics, Springer-Verlag, New York.

van der Waerden, B.L. (**1985**) *A history of algebra*, Springer-Verlag, Berlin-Heidelberg.

Voloch, J. F. (**1987**) A distribuição dos números primos, *Matemática Universitária*, número **06**, 71–82.

Weil, A. (**1987**) *Number theory: an approach through history*, Birkhäuser, Boston.

Weyl, H. (**1982**) *Symmetry*, Princeton University Press, Princeton.

Índice dos Principais Algoritmos

Algoritmo	Objetivo	Página
Algoritmo de divisão	Calcular quociente e resto de uma divisão	20
Algoritmo euclidiano	Calcular máximo divisor comum	24
Algoritmo euclidiano estendido	Calcular o máximo divisor comum e os coeficientes da combinação linear	29/31
Algoritmo da fatoração	determinar o menor fator primo de um inteiro	38
Algoritmo de Fermat	determinar fator de um inteiro	40
Crivo de Eratóstenes	determinar os primos menores que um inteiro dado	64
Teste de Miller	verificar se um inteiro é composto	112
Algoritmo chinês do resto	resolver sistemas de congruencias lineares	120/1
Método de Fermat	determinar fator de um número de Mersenne	157
Método de Euler	achar fator de um número de Fermat	159
Teste de Lucas-Lehmer	primalidade de números de Mersenne	162
Teste de Lucas	determinar se um número é primo	168
Teste de Pepin	determinar se um número de Fermat é primo	169
Teste determinístico	determinar se um número é primo	171
Método de Gauss	achar uma raiz primitiva de $U(p)$	175

Índice dos Principais Resultados

Resultado	Conteúdo	Página
Teorema da divisão	existência e unicidade do quociente e resto	22
Teorema da Fatoração única	todo inteiro é produto de primos de modo único	35
Propriedade Fundamental dos primos	se um primo divide um produto então divide um dos fatores	44
Teorema de inversão	existência de inversos módulo n	83
Princípio de indução finita	método de demonstração	92/8
Teorema de Fermat	a^p é congruente a a módulo p se p é primo	93/5
Teorema de Korselt	caracteriza números de Carmichael	109
Teorema chinês do resto	solução de sistemas de congruencias lineares	122/5
Teorema de Lagrange	a ordem de um subgrupo divide a ordem do grupo	145
Teorema de Euler	$a^{\phi(n)}$ é congruente a 1 módulo n se a e n são primos entre si	148
Lema chave	se $a^k = e$ em um grupo então a ordem de a divide k	156
Teorema da raiz primitiva	se p é primo então o grupo $U(p)$ é cíclico	175

Índice

A

Abel, N. H. 139
Adleman, L. 3
Algarismo 19
Álgebra 19
Algoritmo *17*, 18, 24
  chinês do resto *120*
  da curva elítica 195-196
  da raiz quadrada 199
  das frações continuadas 196
  de divisão 17, 19, *20*, 28
  de fatoração *38*, 39
  de Fermat *veja Fermat*
  euclidiano 17, 23, *24*, 28
  euclidiano estendido 28, *29-30*, 33, 43, 82, 121, 183-184
  de exponenciação *201*
Al Khowarazmi *veja Ben Musa*
Aristóteles 46
Aritmética 8
  modular 69
Array 6, 64
Ars Magna 139
Assinatura eletrônica 3, 189-190
Astronomia 118
Aurélio (dicionário) 17, 91
Axiom *veja Computação algébrica*

B

Bachet, C. G. 9
Base
  de um sistema de numeração 6
  em testes de primalidade *106*
Ben Musa 19
Besicovitch, A. S. 82
Big Bang 39, 93
Binômio de Newton 16, 55, 94
Bloco 180
  invariante 187
Brama, 88, 90, 93

C

C *veja linguagem de programação*
Cálculo 10, 19
Cardano, G. 138-139
Carmichael, R. D. 107
  número de *107*, 108-110, 112, 114-115
César 1,3
Champollion, J.-F. 2
Chave
  de codificação 4, 182, 186
  de decodificação 4, 183, 191
  pública 3, 4, 13, 186
Chocolate 20
Classe de equivalência 70
  da congruência módulo n 73
  e frações 72
Co-fator *24*
Coleta de lixo 7
Computação algébrica 5, 113, 197
  sistemas de 6, 80, 113-114, 184
Congruência módulo n *73*

classes de equivalência da 73
conjunto quociente da 73
é relação de equivalência 73
Congruência módulo um subgrupo 148, 163
Conjectura
de Catalan 12
de Goldbach *veja Goldbach*
dos primos gêmeos 13
Conjunto
com limiar *127*, 130
das partes *71*
quociente *71*, 72, 74
$\mathbb{Z}_n$ *74*
Contagem de freqüência 2, 182
Contra-exemplo 16
Contrapositiva 15, 109
Co-primos *24*
Criptografia 1, 195
de chave pública 3, 186
Criptologia 1
Crivo
de Eratóstenes *veja Eratóstenes*
dos corpos de números 192
quadrático 191
Critérios de divisibilidade 77-79

D

Decifrar 1
Decodificar 1, 183
Demonstração 14
por contradição 44, 50, 59
por redução ao absurdo 44, 46, 59
por indução finita 92-95
Derive *veja Computação algébrica*
Diffie, W. 3
Diofanto 8, 48, 80-81
Disquisitiones Arithmeticæ *veja Gauss*
Distributividade 77
Divisor *23*
Dominó 92

E

Elemento
inversível módulo n 82-83
simétrico 77
Elementos *veja Euclides*
El Gamal 191
Equação
diofantina 80-82, 85
linear módulo n 84, 117-118
polinomial 96, 139, 196
teoria das 138
Eratóstenes 62
Crivo de 53, 62-63, *64*, 66, 67, 103
Euclides 8, 14, 16-17, 24, 44-46, 49, 60
Euler, L. 10, 12, 50, 57, 60, 67, 81, 91, 102, 158
teorema de 148, 152, 185
método de 159, 165

F

Fator *24*
Fermat, Pierre de 9, 12, 40, 47, 56, 58, 86, 93-96, 155, 158
algoritmo de *40*, 41-43, 51, 187, 195
método de *157*, 165
número de *56*, 57-58, 66, 85, 158, 168-169
pequeno teorema *veja Fermat, teorema de*
primos de 58, 160
Samuel 10
teorema de 87, *93*, 94, *95*, 96, 101, 105, 125, 148, 156, 159, 175, 177, 185
último teorema de 10, 47, 80, 103
Ferrari, L. 139
Ferro, Scipione del 138
Fibonacci (seqüência de) 26, 32
Fior, A. M. 138
Física 134

Forma reduzida
de uma fração 45
módulo n 74, 183
Franklin, B. 126
Frenicle, B. 57-58, 158, 165
Função
π 61
ϕ de Euler *141*, 142-144, 186
totiente 141

G
Galois, E. 139-140, 196
teoria de 196
Gauss, C. F. 11, 19, 36, 140, 160, 175, 179
método de 175-178
Geometria 8, 10, 11, 13, 134, 139
GIMPS (Great Internet Mersenne Prime Search) 165
Goldbach, C. 10, 12
conjectura de 12
Grupo *133*, 163
abeliano *134*, 137, 163, 173, 179
cíclico 167
diedral *138*
gerador de um *146*, 165
ordem de um *veja ordem*
simples 197
teoria de 58, 87, 133, 196, 197

H
Hacker 189
Hadamard, J. 61-62
Hajratwala, N. 165
Hardy, G. H. 36
Hellman, M. E. 3
Heródoto 13, 45
Hieróglifos 2
Hipótese 14
de indução *93*, 95, 97, 98, 99

I
Indução 91
finita 87, 91-93, 100
princípio de *92*, *98*
Internet 3
Inverso 82
módulo n 82
em um grupo 134

K
Knuth, D. E. 28
Korselt, A. 108
teorema de 109, 169

L
Laço 20
Lehmer, D. H. 161
Leibniz, G. W. 19, 106
Lema 26
chave 156, 158, 164, 168, 171, 173, 176, 177
Linguagem de programação 5, 161
Liouville, J. 140
Lista 6
Littlewood, J. E. 68
Logaritmo
discreto 192
em base dez 166
em base e 62, 67, 187, 188
Logística 8
Lucas, E. 161

M
Manual de Aritmética do Mestre Sun 120
Maple *veja Computação algébrica*
Máquina
de calcular 22, 74, 92
de Turing 3
Mathematica *veja Computação algébrica*

Máximo divisor comum 8, 17, 23, *24*, 26, 82
Merkle, R. C. 3
Mersenne, M. 9-10, 56-57, 155
  número de *56*, 57, 155, 161-165
  primos de 56, 57, 157
Miller, G. L. 111
  teste de 111, *112*, 113-114, 116, 168, 172, 188
Mínimo múltiplo comum *33*, 49, 173, 176
Módulo 73
Múltiplo *24*

N

Netscape 3
Nickel, L. 165
Nicômaco de Gerasa 62, 120
Noll, C. 165
Número
  altamente composto *50*
  composto 5, 35
  de Fermat *veja Fermat*
  de Mersenne *veja Mersenne*
  hexagonal *100*
  irracional 44, 162
  perfeito 8, *49*, 50, 56, 165
  primo *veja primo*
  teoria dos 1, 4, 8, 61, 195

O

Operação
  módulo n 75-77
  de um grupo 133
Ordem
  de um grupo *134*
  de um elemento *146*

P

Paraty 181
Pascal, B. 10, 56, 92
  linguagem *veja linguagem de programação*
Pepin, J. F. T. 168
  teste de 169, 180
Pitágoras 46
Platão 46
Polígono 70
  regular 138, 160
Polinômio
  divisão de 99, 197
  irredutível 197
  fatoração de um 197
Potência
  em um grupo 145
  módulo n 79-80, 125-126
Pré-codificação 181
Primo 4, 8, 35, 87, 101
  da forma $4n + 1$ 66-68, 101
  da forma $4n + 3$ 66-68, 86, 102-103
  de Fermat *veja Fermat*
  de Mersenne *veja Mersenne*
  distribuição dos 61-62, 128, 187
  entre si *24*
  fórmula exponencial para 56-58
  fórmula fatorial para 59-60
  fórmula polinomial para 53-56
  infinidade dos 60, 66
Princípio
  de indução finita *veja indução*
  de superposição 198
Probabilidade 10, 34
Produto vetorial 133-134
Progressão geométrica 8, 16, 49
Propriedade
  da adição módulo n 76
  da multiplicação módulo n 77
  fundamental dos primos *44*, 45, 47
Pseudoprimo 106, 152
  da forma p^2 116, 152
  forte 113, 114, 115

para base 2 106
para base 2 e 3 106
para todas as bases 107

Q

Quântico
computador 198
mecânica 198

R

Rabin, M. 192
método de 192
Raiz primitiva 83, 175, 176, 178
teorema da 110, 167, 175, 176
Ramanujan, S. 51
Reciprocidade quadrática 169
Recorrência 90, 93
Reflexão 135-138, 150, 175
Relação *69*, 85
de equivalência 69, *70*, 85, 148
reflexiva *70*, 85, 149
simétrica *70*, 85, 149
transitiva *70*, 85, 149
Renascença 9, 138
Riemann, B. 61
Rivest, R. L. 3
Rotação 135-138, 150, 175
RSA 1, 3, 13, 32, 39, 53, 181-190
195-196
chave de codificação do 182, 186, 190
chave de decodificação do 183, 190
codificação no 182-183
decodificação no 183
parâmetros do 4, 182, 184
segurança do 186

S

Shamir, A. 3
Simetria 134-135, 197
de um polígono regular 138
de um quadrado 150
de um triângulo eqüilátero 135-137, 175
Sistemas
de computação algébrica
veja computação algébrica
de congruências lineares 118-127
Subgrupo *144*, 147-148
cíclico 146
de ordem prima 146
próprio 147
Supercomputador 157, 160
Shor, P. 198

T

Tabela
de um grupo 137
do teorema chinês do resto 122-123, 142
Tartaglia, N. 138
Taylor, R. 48
Teorema 14, 18, 31
chinês do resto *122*, 123-125, 127, 142
da fatoração única 23, *35*, 60
da infinidade dos primos *60*
da raiz primitiva *veja raiz primitiva*
de divisão *22*
de Euler *veja Euler*
de Fermat *veja Fermat*
de inversão *83*, 95, 107
de Korselt *veja Korselt*
de Lagrange 145, 148-150, 156, 164
de Wilson *179*
dos números primos *62*, 67
Testemunha 105
Teste
de Lucas 168
de Lucas-Lehmer, 161-165
de Miller *veja Miller*
de primalidade 168, 188
The Art of computer programming *veja Knuth*

Toro 123
Torres de Hanói 87, 92
Turing, A. 3

V

Vallé-Poussin, C. J. de 62
Vetor 6, 64, 133

W

Weyl, H. 197
Wiles, A. 48
Wright, E. M. 36

Série de Computação e Matemática

Números Inteiros e Criptografia RSA

A palavra criptografia ainda evoca imagens de agentes secretos sorrateiramente transferindo informações sigilosas a nações rivais. Entretanto, a principal missão da moderna criptografia é proteger as informações referentes a transações bancárias e comerciais que transitam entre computadores numa rede.

Este livro é uma introdução elementar a um dos métodos criptográficos mais populares atualmente, o RSA, e à área da matemática que lhe serve de fundamento, a teoria de números.

O tratamento matemático do material é rigoroso, mas a apresentação é pouco convencional, recheada de exemplos concretos e notas históricas. Isto torna o livro mais agradável de ler do que o compêndio matemático tradicional.

O enfoque adotado é francamente algorítmico, e as demonstrações são, sempre que possível, construtivas. Além disso os pré-requisitos necessários à leitura do livro são mínimos, o que o põe ao alcance de um estudante do final do segundo grau. O material do livro tem sido utilizado em um curso de álgebra para informática na UFRJ.

Uma publicação:

impa