Lista de Exercícios S. C. Coutinho

Capítulo 1 Algoritmos Fundamentais

1. Para cada par a, b de números inteiros abaixo calcule o mdc e determine α e β de modo que $mdc(a,b) = \alpha \cdot a + \beta \cdot b$. Utilize qualquer linguagem de programação para implementar computacionalmente a solução para esses problemas.

   a) 14 e 35

   b) 252 e 180

   c) 6643 e 2873

   d) 272828282 e 3242

2. 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 $d = mdc(a,b)$. Então existem inteiros a' e b' tais que $a = da'$ e $b = db'$. Logo

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

Teorema 1: A equação $ax + by = c$ só pode ter solução se d dividir c. Demonstrar este teorema não será necessário nesta questão.

Teorema 2: Se o teorema 1 é verificado ($c = dc'$) e a equação reduzida é construída na forma $a'x + b'y = c'$ então, qualquer solução da equação original é solução da reduzida e vice-versa.

Teorema 3: Podemos utilizar o algoritmo euclidiano estendido para achar α e β inteiros tais que $\alpha \cdot a + \beta \cdot b = 1$ e neste caso a equação reduzida tem soluções $x = c'\alpha$ e $y = c'\beta$.

Assim, escreva um programa para determinar uma solução inteira para a equação

$ax+by=c$, tendo como entradas 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. Assim, o código desenvolvido na equação 1 poderá ser utilizado e adaptado para resolver essa questão 2.

3. Escreva um programa que, tendo como entrada dois inteiros $\underline{a}$ e $\underline{b}$, determina o máximo divisor comum de $\underline{a}$ e $\underline{b}$. Adapte o seu programa para gerar aleatoriamente pares de inteiros $\underline{a}$ e $\underline{b}$ e calcular o $mdc(a,b)$. O programa deve ter como entrada o número total de pares que você deseja testar, e como saída o quociente

$$\boxed{\frac{\text{total de pares cujo mdc é 1}}{\text{total de pares testados}} \cong \frac{6}{\pi^2}}$$

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 (pares), 100 (pares), 1000 (pares), 10.000 (pares) e 100.000 (pares) 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 $6/\pi^2$. Como é que este valor se compara aos resultados experimentais que você obteve?

S. C. Coutinho

Capítulo 2 — Fatoração Única

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

2. Faça um programa de computador para os algoritmos 3 (fatoração convencional) e 4 (fatoração de Fermat) e compare o desempenho computacional de ambos utilizando os exemplos numéricos da questão 1. (*1)

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

Capítulo 3 — Números Primos

1. Ache, procedendo por tentativa, a fatoração em primos de $p^{\#}+1$ quando:
   (1) $p = 17$
   (2) $p = 13$

2. Implemente computacionalmente o crivo de Eratóstenes para geração de números primos. Tente explorar toda a capacidade de memória do seu computador e tempo máximo de processamento para obter o máximo possível de números primos. Não vale baixar da Internet listas de números primos já prontas. Coloque nos Apêndices a listagem de seu código computacional.

3. Procure na Internet e descreva matematicamente o método de Fermat para fatorar números de Mersenne. Não é necessário implementar este método computacionalmente. Basicamente, este método permite encontrar fatores primos para $M(p)$, quando p é primo.

(*1) O Exercício 2 do Capítulo 2 é o primeiro de uma seqüência que termina com o exercício 8 do Capítulo 11!

S. C. Coutinho

4. Exercício opcional (não é obrigatório). 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 $x/\ln x$ 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:

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

onde ln 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 10.000. Compare com os valores correspondentes de $\pi(x) - x/\ln x$. O que você conclui da análise destas tabelas? Marque nesta tabela o tempo de processamento exigido pelo Crivo de Eratóstenes.

Nota: Se $x = 10^{16}$, então $\underbrace{\pi(x) - \left[\frac{x}{\ln x}\right]}_{\text{deveria dar zero, pois } \lim_{x\to\infty} \frac{\pi(x)\ln x}{x} = 1} = 7.804.289.844.393$

S. C. Coutinho

Capítulo 4 | Aritmética Modular

(1.) Vimos que $a \equiv a_2 \cdot 10^2 + a_1 \cdot 10 + a_0 \equiv 2a_2 + 3a_1 + a_0 \pmod 7$. Dando continuidade a isso, encontre algoritmos eficientes que testem a divisibilidade de um número $\underline{a}$ qualquer por 7 nos seguintes casos:

a) $a = a_1 a_2 a_3 a_4 a_5 a_6$ (Utilize o Teorema 4 da pg. 63)

b) $a = a_1 a_2 \cdots a_{12}$ (Teorema 4 da pg. 63). Teste com um exemplo!

(2.) Prove que para $a = a_m \cdot 10^m + a_{m-1} \cdot 10^{m-1} + \cdots + a_1 \cdot 10 + a_0$ temos que:

$$a \equiv a_m (-1)^m + a_{m-1}(-1)^{m-1} + \cdots + a_2 - a_1 + a_0 \pmod{11}$$

Nota: utilize o teorema 4 para provar isto (pg. 63). Assim, um número qualquer $\underline{a}$ é divisível por 11 se, e somente se, a <u>soma alternada dos seus algarismos</u> é também divisível por 11. Por exemplo, 3443 é divisível por 11, já que $3-4+4-3=0$ que é divisível por 11.

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

(4.) Quais os elementos de $\mathbb{Z}_4$ que têm inversos? E de $\mathbb{Z}_8$?

(5.) Resolva as seguintes equações:

a) $4x \equiv 3 \pmod 4$.

b) (optativo) $3x + 2 \equiv 0 \pmod 4$.

(6.) Implemente um programa que calcule potências módulo m que é descrito no Apêndice A. O algoritmo deverá ter como entrada $\underline{a}$, $\underline{k}$ e $\underline{m}$, onde $\underline{a}$ é um inteiro qualquer e $\underline{k}$ e $\underline{m}$ são inteiros positivos. A saída deverá ser a <u>forma reduzida</u> de $\underline{a}^k$ módulo $\underline{m}$. Este algoritmo é uma <u>parte fundamental</u> de <u>quase todos os algoritmos</u> que estudaremos <u>a partir do Capítulo 6</u>.

(optativo)

7. Mostre que $p = 274177 = 1071 \cdot 2^8 + 1$ é fator primo do número de Fermat $F(6)$. Lembre-se que $F(6) = 2^{2^6} + 1$ pois, $F(n) = 2^{2^n} + 1$ (por definição).

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!

8. Utilize o algoritmo da página 76 para calcular P tal que:

$$\boxed{5^3 \equiv P \pmod 3}$$

Sugestão: Siga o exemplo da página 76.

Capítulo 5 Indução de Fermat

1. Prove por indução que:

(1) $n^3 + 2n$ é divisível por 3 para todo inteiro $n \geq 1$.

2. (optativo) 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.

3. Mostre, usando o Teorema de Fermat, que $2^{70} + 3^{70}$ é divisível por 13.

4. Calcule o resto da divisão de:

(1) $39^{50!}$ por 2251;

(2) (optativo) 19^{39^4} por 191.

5. (optativo) 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 $\pm a^{k+1}$.

6. 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 5 acima 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. (*2)

(*2) Este exercício é o segundo de uma seqüência que termina com o exercício 8 do capítulo 11!

S. C. Coutinho

Capítulo 6 | Pseudoprimos

(1) Testando a eficácia do teste de Leibniz. Faça um programa de computador que tenha como entradas um número inteiro positivo m e uma base inteira positiva tal que $1 < b < m-1$. Em seguida, este programa verifica se todos os testes abaixo são simultaneamente verdadeiros:

$$2^{m-1} \equiv 1 \pmod m$$
$$3^{m-1} \equiv 1 \pmod m$$
$$4^{m-1} \equiv 1 \pmod m$$
$$\vdots$$
$$(b-1)^{m-1} \equiv 1 \pmod m$$
$$b^{m-1} \equiv 1 \pmod m$$

Por exemplo, teste este programa para $m = 343$ e $b = 5$, ou seja, o programa terá que verificar se:

$$2^{342} \equiv 1 \pmod{343}$$
$$3^{342} \equiv 1 \pmod{343}$$
$$4^{342} \equiv 1 \pmod{343}$$
$$5^{342} \equiv 1 \pmod{343}$$

Assim, se o valor de $m = 343$ passar simultaneamente nestes quatro testes é plausível afirmar que $m = 343$ é primo? Faça o mesmo para $m = 347$ e $b = 10$.

Nota: para verificar cada uma das quatro congruências dadas acima utilizar o algoritmo da página 76 da apóstila (Apêndice A).

2. Números de Carmichael. Sabemos que $m = 561$ é o menor número de Carmichael. Verifique isto através da definição. Para isto faça o seguinte:

$$b^{m-1} \equiv 1 \pmod{m} \qquad 1 < b < m-1$$

Passo 1:

$$\left.\begin{array}{l} 2^{560} \equiv 1 \pmod{561} \\ 3^{560} \equiv 1 \pmod{561} \\ 4^{560} \equiv 1 \pmod{561} \\ \vdots \\ 559^{560} \equiv 1 \pmod{561} \end{array}\right\}$$

Verifique quais destas congruências são satisfeitas utilizando o algoritmo da página 76 da apostila (Apêndice A)

Passo 2: Separe todos os valores de b onde $1 < b < m-1$ onde as congruências são verificadas. Digamos que esse conjunto seja $\{b_1, b_2, \ldots, b_q\}$.

Passo 3: Verifique que $\mathrm{mdc}(b_i, m) = 1$ para $i = 1, 2, \ldots, q$.

Nota: tanto as questões 1 e 2 necessitam de um software para serem resolvidas. É completamente impossível resolvê-las com lápis e papel.

3. Quais dos seguintes números são pseudoprimos para a base 2: 645, 567 e 701? Quais são pseudoprimos para a base 3? Quais são primos?

4. Fatore 29341 e mostre que é um número de Carmichael. Evidentemente, utilize para isto o Teorema de Korselt.

5. 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?

(Optativo)

6. Escreva um programa de computador 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?

7. (optativo) Escreva um programa de computador que implemente e teste o Teorema de Korselt. Para isso será necessário aplicar algum algoritmo de fatoração explicado no Capítulo 2. Desta forma, dado um número inteiro positivo ímpar m o programa deverá:

(i) Fatorar m;

(ii) Aplicar o Teste de Korselt para verificar se m é um número de Carmichael.

Exemplos de Números que são de Carmichael:

$561 = 3 \times 11 \times 17$

$1105 = 5 \times 13 \times 17$

$1729 = 7 \times 13 \times 19$

$2465 = 5 \times 17 \times 29$

$2821 = 7 \times 13 \times 31$

$6601 = 7 \times 23 \times 41$

$8911 = 7 \times 19 \times 67$

$10585 = 5 \times 29 \times 73$

$101101 = 7 \times 11 \times 13 \times 101$

$115921 = 13 \times 37 \times 241$

$252601 = 41 \times 61 \times 101$

S.C. Coutinho

$\boxed{\text{Capítulo 7}}$ Sistemas de Congruências

1. Um problema de Yih-hing (717 d.C.): ache a solução do sistema:

$$x \equiv 1 \pmod{2}$$
$$x \equiv 2 \pmod{5}$$
$$x \equiv 5 \pmod{12}$$

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 o 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.(*) Sejam p e q primos distintos e $n = p.q$. Suponhamos que ambos os primos deixam resto 3 na divisão por 4. Escreva um programa que, tendo como entrada p, q e a, calcule uma solução de $x^2 \equiv a \pmod{p}$ e $x^2 \equiv a \pmod{q}$; veja o exercício 6. do Capítulo 5.

(*) Este exercício é o terceiro de uma seqüência que termina com o exercício 8 do Capítulo 11.

4. Seja $\mathcal{L} = \{11, 13, 17, 19, 23\}$

$N = 11 \cdot 13 = 143$

$M = 23$

$k = 2$

$s = 30$ (senha)

$S = \{(11,19), (13,17), (17,13), (19,11), (23,7)\}$

- Recupere a senha s nos seguintes casos:

a) Quando os funcionários que estão no banco são (19, 11) e (13, 17);

b) Quando o funcionário que está no banco é somente (11, 19);

c) Quando os funcionários que estão no banco são (11, 19), (17, 13) e (19, 11).

- Descreva em linguagem algoritmo como determinar o conjunto S acima conhecendo-se somente $\mathcal{L}$ e k. Observe que, devido ao teorema do resto chinês, $\mathcal{L}$ só pode possuir números primos. O número de elementos de $\mathcal{L}$ é m. Assim, m será a quantidade de pessoas do banco que receberão senhas distintas (na verdade pares de senha) e k é o número mínimo de pessoas para recuperar a senha facilmente.
Por que se o número de pessoas é menor que k, recuperar a senha é difícil?

5. Estude a teoria dos grupos no livro de Scheinerman de Matemática Discreta (pgs. 395-451). Concentre seus estudos na criptografia RSA somente (pgs. 444-448). Feito isto explique o funcionamento da criptografia RSA e gere um algoritmo computacional para codificar a informação e outro para decodificar a informação. Faça o mesmo para a criptografia de Rabin (pg. 437-444). Motivacional! Vejam que são poucas páginas...