UNIVERSIDADE DE SÃO PAULO

TRABALHO DE FORMATURA

Teoria dos Números e Computação: Uma abordagem utilizando problemas de competições de programação

Autor:
Antonio R. de Campos Junior

Supervisor:
Dr. Carlos Eduardo Ferreira

Tese apresentada em cumprimento dos requisitos para o curso Bacharel em Ciência da Computação

Instituto de Matemática e Estatística

18 de outubro de 2015

"To raise new questions, new possibilities, to regard old problems from a new angle, requires creative imagination and marks real advance in science."

Albert Einstein

Resumo

Teoria do Números é um vasto ramo da matemática que estuda números inteiros. Números primos, fatorização de números inteiros, funções aritméticas, são alguns dos tópicos mais estudados e também importantes para resolução de problemas computacionais.

Hoje em dia a importância da Teoria do Números na Computação é inquestionável, e desse modo, esse trabalho vem ilustrar como a teoria pode ser aplicada na criação de algoritmos para resolução de problemas computacionais, em especial problemas de competições de programação.

Equações diofantinas, Congruência Modular, Números de Fibonacci, são alguns dos assuntos que serão abordados nesse trabalho. Após a devida demostração da teoria serão exibidos alguns problemas de competições de programação que aplicam essa teoria, seguido da implementação e análise do algoritmo que resolve o problema abordado.

Agradecimentos

I like to acknowledge ...

Sumário

Lista de Figuras

Lista de Tabelas

For/Dedicated to/To my…

Capítulo 1

Divisibilidade

1.1 Introdução

Nesse seção vamos descrever algumas definições e propriedades dos números inteiros que serão utilizados ao longo desse trabalho.

Definição 1

Corolário 1 *Dado um subconjunto dos inteiros $S = \{S_1, S_2, S_3, ..., S_n\}$ ordenado crescentemente, e um número inteiro d, tal que, $d|(S_i - S_{i-1})$, $2 \leq i \leq n$.*

Nessas Condições temos que: $d|(S_i - S_j)$, $\forall S_i, S_j \in S$.

Demonstração: Tome $S_i, S_j \in S$ quaisquer, e sem perda de generalidade assuma que $S_i \geq S_j$ (ie, $i \geq j$, pois S está ordenado crescentemente).

Como $i \geq j$, tome $r \in \mathbb{N}$ como sendo a diferença entre i e $j : i = j + r$.

Vamos agora provar por indução que $d|(S_{j+r} - S_j)$.

Para $r = 0$ ou $r = 1$ a demostração segue trivialmente.

Assuma que o corolário funciona para $(r - 1)$, ie, $d|(S_{j+r-1} - S_j)$.

Temos então que:

$d|(S_{j+r} - S_{j+r-1}) \Rightarrow d|(S_{j+r} - S_{j+r-1}) + (S_{j+r-1} - S_j) \Rightarrow d|(S_{j+r} - S_j)$

Corolário 2 *O **Corolário** 1 funciona mesmo se o conjunto S não estiver ordenado.*

Demonstração: Deixaremos a demostração a cargo do leitor.

Teorema 1 *Dados dois inteiros quaisquer a e b, com $b > 0$, então existe um único par q e r tal que:*

$a = qb + r$, com $0 \leq r < b$

Demonstração: Deixaremos a demostração a cargo do leitor.

1.2 Números Primos

Definição 2 *Todo número inteiro n (n > 1) que têm apenas dois divisores distintos (1 e n) é chamado de número primo. Se n (n > 1) não for primo, dizemos que n é número composto.*

1.3 Máximo Divisor Comum

Definição 3 *O Máximo Divisor Comum de dois inteiros quaisquer a e b (com a ou b diferente de zero), denotado por $MDC(a, b)$, é o maior inteiro que divide ambos a e b.*

Algorithm 1 Máximo Divisor Comum

```
procedure MMC (a, b)
    while b ≠ 0 do
        t ← b
        b ← a mod b
        a ← t
    return a
```

Pseudocódigo:

Corolário 3 $MDC(a, b) = d \Rightarrow MDC(a/d, b/d) = 1$

Demonstração:

Teorema 2 (Teorema de Bézout) $\forall\, a, b \in \mathbb{Z}, \exists\, x, y \in \mathbb{Z} \mid ax + by = mdc(a, b)$.

Demonstração: De acordo com Teorema 2

1.4 Crivo de Erastóteles

1.5 Problemas Propostos

1.5.1 UVA-10407

10407 - Simple Division

Resumo: Tome $P(S) := \{x \in \mathbb{Z} \mid \forall a, b \in S, a \equiv b(mod\ x)\}$ em que $S \subset \mathbb{Z}$.
O problema consiste em encontrar o valor máximo de $P(S)$ dado um conjunto S.

Solução: Seja $S = \{S_1, S_2, S_3, ..., S_n\}$, com $n = |S|$, o conjunto dado pelo problema (assumiremos que os valores de S estão ordenados crescentemente).
Tome um número qualquer $d \in P(S)$. Por definição temos que $\forall S_i, S_j \in S$, $S_i \equiv S_j(mod\ d) \Rightarrow (S_i - S_j) \equiv 0(mod\ d) \Rightarrow d \mid (S_i - S_j)$.
Pelo **Corolário** 1 sabemos que:
$d|(S_i - S_{i-1}), \forall i \in \mathbb{N}, 2 \leq i \leq n \Rightarrow d|(S_i - S_j), \forall S_i, S_j \in S \Rightarrow d \in P(S)$.
E desse modo, para calcular o valor máximo de $P(S)$ só precisamos calcular o Máximo Divisor Comum das diferenças $(S_i - S_{i-1})$ com i variando de 2 à n $\square$.

Pseudocódigo:

Algorithm 2 Simple Division

```
procedure GetMaximumValue (S)
    S ← sort(S)                         ▷ sort(X) retorna o conjunto X ordenado.
    maxValue ← 0
    for i := 2 to |S| do
        maxValue ← MDC(maxValue, S_i − S_{i−1})
    return maxValue
```

Capítulo 2

Congruência

2.1 Congruência

2.2 Congruência Linear

2.3 Teorema de Fermat, Euler e Wilson

2.4 Teorema do Resto Chinês

Teorema 3 (Teorema do Resto Chinês) *Tome o sistema de congruências lineares:*

$a_1x \equiv c_1(mod\, m_1)$
$a_2x \equiv c_2(mod\, m_2)$
$a_3x \equiv c_3(mod\, m_3)$
$...$
$a_nx \equiv c_n(mod\, m_n)$

Em que $c_i \in \mathbb{Z}$, $MDC(a_i, m_i) = 1$, e $MDC(m_i, m_j) = 1$ para $i \neq j$ Nessas condições o sistema acima tem solução única módulo M, em que $M = m_1m_2m_3...m_n$.

Demonstração: Deixaremos a demostração a cargo do leitor.

2.5 Problemas Propostos

Capítulo 3

Funções Aritméticas

3.1 φ de Euler

Definição 4 *A Função Totiente de Euler, denotada por $\varphi(n)$, é a função aritmética que conta o número de inteiros positivos menores ou iguais a n que são primos entre si com n.*
$\varphi(n) := |\{x \in \mathbb{N}^* \mid MDC(x, n) = 1\}|$

Teorema 4 *$\varphi(n^k) = n^{k-1}\varphi(n)$, para inteiros positiovos quaisquer n e k. Em particular $\varphi(p^k) = (p^k - p^{k-1})$, para p primo.*

Demonstração:

Teorema 5 *$\varphi(n)$ é função multiplicativa, ie, $\varphi(mn) = \varphi(m)\varphi(n)$ para $MDC(m, n) = 1$.*

Demonstração:

Teorema 6 (Fórmula Produto de Euler) $\varphi(n) = n\prod_{p|n}(1 - \frac{1}{p})$

Demonstração: Pelo Teorema X, Teorema 4, Teorema 5 segue as seguintes recorrências:
$\varphi(n) = \varphi(p_1^{a_1}p_2^{a_2}...p_k^{a_k})$
$\varphi(n) = \varphi(p_1^{a_1})\varphi(p_2^{a_2})...\varphi(p_k^{a_k})$
$\varphi(n) = (p_1^{a_1} - p_1^{a_1-1})(p_2^{a_2} - p_2^{a_2-1})...(p_k^{a_k} - p_k^{a_k-1})$
$\varphi(n) = p_1^{a_1}p_2^{a_2}...p_k^{a_k}(1 - 1/p_1)(1 - 1/p_2)...(1 - 1/p_k)$
$\varphi(n) = n\prod_{p|n}(1 - \frac{1}{p})$ $\square$

3.2 Sequência de Fibonacci

Definição 5

3.3 Problemas Propostos

3.3.1 UVA-11424

11424 - GCD - Extreme (I)

Resumo: É dado um inteiro positivo N ($1 < N < 200001$). O problema consiste em calcular o mais rápido possível a expressão:
$G(N) = \sum_{i=1}^{N-1}\sum_{j=i+1}^{N} MDC(i, j)$.

Solução: Trivialmente a expressão acima pode ser calculada em tempo proporcional à $O(n^2 log(N))$, porém essa solução consome muito tempo e não será aceita no

Judge Online. Vamos então mostrar uma solução mais eficiente.

Primeiramente reescrevemos a expressão acima da seguinte maneira:
$G(N) = \sum_{j=2}^{N} \sum_{i=1}^{j-1} MDC(i,j)$ ($\triangleright$ Observe que as expressão são equivalentes).
Tome agora a função $F(M) = \sum_{i=1}^{M-1} MDC(i,M) \Rightarrow G(N) = \sum_{j=2}^{N} F(j)$.
Sabemos que todos os valores resultantes do método $MDC(i,M)$ calculados em $F(M)$ são divisores de M. Desse modo, podemos reescrever $F(M)$ da seguinte maneira:
$F(M) = \sum_{i=1}^{M-1} MDC(i,M) = \sum_{l=1}^{n} \lambda_l d_l$, em que, $d_1, d_2, ..., d_n$ são os divisores de M, λ_l é o número de vezes que o divisor d_l aparece na somatória $\sum_{i=1}^{M-1} MDC(i,N)$, e n é o número de divisores de M.

Pelo Corolario 3 temos que: $MDC(i,M) = d_l \Rightarrow MDC(i/d_l, M/d_l) = 1$. Logo o número de vezes que o divisor d_l aparece na somatória, será igual ao número de primos entre si com (M/d_l), ie, $\lambda_l = \varphi(M/d_l)$.
Reescrevendo novamente $F(M)$, temos:
$F(M) = \sum_{i=1}^{M-1} MDC(i,M) = \sum_{l=1}^{n} \lambda_l d_l = \sum_{l=1}^{n} \varphi(M/d_l) d_l$.
$G(N) = \sum_{j=2}^{N} \sum_{l=1}^{n} \varphi(j/d_l) d_l \;\square$.

Pseudocódigo:

Algorithm 3 GCD - Etreme(I)

```
1: procedure G (N)
2:     φ[] ← PHI(N)
3:     solution ← 0
4:     for j := 2 to N do
5:         for each divisor d de j do
6:             solution ← solution + φ[j/d]d
7:     return solution
```

Análise: O método $PHI(N)$ na linha 2 consome tempo proporcional à $O(N\sqrt{N})$.
O número de divisores de j é proporcional à $O(\sqrt{N})$, já que $j \leq N$.
Assim a complexidade das linhas 4, 5, 6 do algoritmo é $O(N\sqrt{N})$.
Complexidade final do algoritmo: $O(N\sqrt{N})$.
OBS.: Para resolver o problema no Judge Online será preciso armazenar as soluções usando Programação Dinâmica.

3.3.2 TJU-3506

3506 - Euler Function

Resumo: É dado dois números positivos n, m ($1 < n < 10^7$, $1 < m < 10^9$). O problenas consiste em calcular a expressão: $\varphi(n^m) \bmod 201004$.

Solução: Pelo Teorema 4

Pseudocódigo:
Análise:

Algorithm 4 GCD - Etreme(I)

```
procedure G (N)
    φ[] ← PHI(N)
    solutuion ← 0
    for j := 2 to N do
        for each divisor d de j do
            solution ← solution + φ[j/d]d
    return solutuion
```

3.3.3 CodeChef-MODEFB

71544 - Another Fibonacci

Resumo: É dado dois números inteiros N, K ($1 \leq N \leq 50000$, $1 \leq K \leq N$) e um conjunto $S \subset \mathbb{N}$ com N elementos, tal que, $\forall s \in S, 1 \leq s \leq 10^9$. O problenas consiste em calcular a expressão: $F(S) = \sum_{A \subset S \ e \ |A|=K} Fib(sum(A))$, onde $sum(A) = \sum_{a \in A} a$.

Solução:

Pseudocódigo:

Algorithm 5 Another Fibonacci

```
procedure F (S)
```

Análise:

3.3.4 UVA-10311

10311 - Goldbach and Euler

Resumo: É dado um número inteiro n ($0 < n \leq 10^8$). O problema consite em verificar se n pode, ou não pode, ser escrito como a soma de dois números primos. E em caso afirmativo encontrar o valor desses dois primos.

Solução:

Pseudocódigo:

Algorithm 6 Another Fibonacci

```
procedure F (S)
```

Análise:

Capítulo 4

Conclusão

...

Apêndice A

Curiosidades da ACM-ICPC

ACM-ICPC (International Collegiate Programming Contest) é uma competição de programação de várias etapas e baseada em equipe. O principal objetivo é encontrar algoritmos eficientes, que resolvem os problemas abordados pela competição, o mais rápido possível.

Nos últimos anos a ACM-ICPC teve um crescimento significativo. Se compararmos o número de competidores, temos que de 1997 (ano em que começou o patrocinio da IBM) até 2014 houve um aumento maior que 1500%, totalizando 38160 competidores de 2534 universidades em 101 países ao redor do mundo.

Para mais informações sobre as competições passadas acesse icpc.baylor.edu.

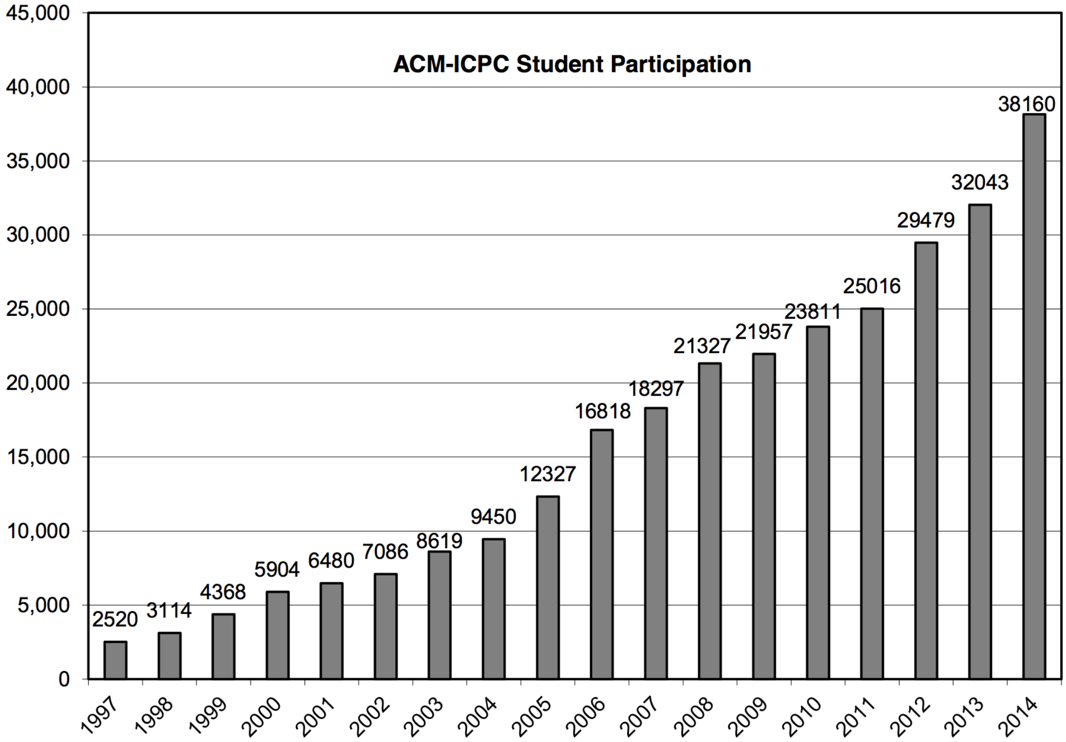

FIGURA A.1: Crescimento do número de participantes por ano.

Apêndice B

Juízes Online (Online Judges)

Online Judges são plataformas online que contam com um banco de dados com diversos tipos de problemas de competições de programação, e com um sistema de correção online.

Para afirmar que sua solução está correta, basta enviar o código fonte da sua solução (em geral escrito em C++ ou JAVA) para uma dessas plataformas.

Alguns desses Online Judges são citados em seguida.

B.1 UVa

Criado em 1995 pelo matemático Miguel Ángel Revilla, é atualmente um dos Online Judges mais famoso entre os participantes da ACM-ICPC.

É hospedado pela Universidade de Valhadolide e conta com mais de 100000 usuários registrados.

Site: https://uva.onlinejudge.org/

B.2 Topcoder

Empresa que administra competições de programação nas linguagens Java, C++ e C#.

É responsável também por aplicar competições de design e desenvolvimento de software.

Site: https://www.topcoder.com/

B.3 Codeforces

Site Russo dedicado competições de programação.

Em 2013, Codeforces superou Topcoder com relação ao número de usuários ativos, apesar de ter sido criado quase 10 anos depois.

O estilo de problemas que esse site aplica é similar aos problemas encontrados na ACM-ICPC.

Site: http://codeforces.com/

B.4 CodeChef

Iniciativa educacional sem fins lucrativos lançada em 2009 pela Direct.

É uma plataforma de progamação competitiva que suporta mais de 35 linguagens de programação.

Site: https://www.codechef.com/

Bibliografia

Arnold, A. S. et al. (1998). "A Simple Extended-Cavity Diode Laser". Em: *Review of Scientific Instruments* 69.3, pp. 1236–1239. URL: http://link.aip.org/link/?RSI/69/1236/1.

Hawthorn, C. J., K. P. Weber e R. E. Scholten (2001). "Littrow Configuration Tunable External Cavity Diode Laser with Fixed Direction Output Beam". Em: *Review of Scientific Instruments* 72.12, pp. 4477–4479. URL: http://link.aip.org/link/?RSI/72/4477/1.

Wieman, Carl E. e Leo Hollberg (1991). "Using Diode Lasers for Atomic Physics". Em: *Review of Scientific Instruments* 62.1, pp. 1–20. URL: http://link.aip.org/link/?RSI/62/1/1.