M342 Álgebra Computacional

Christian Lomp

FCUP

10 de Outubro de 2011

Equações Diofantinas Lineares

A equação diofantinas lineares:

$$a_1x_1 + a_2x_2 + \ldots + a_nx_n = c,$$

com $a_1, \ldots, a_n, c \in \mathbb{Z}$ tem soluções $(x_1, \ldots, x_n) \in \mathbb{Z}^n$ se e só se $\mathrm{mdc}(a_1, \ldots, a_n)|c$.

Equações Diofantinas Lineares

$$ax + by = c \qquad \Leftrightarrow \qquad a'x + b'y = c'$$

onde $a' = \frac{a}{\mathrm{mdc}(a,b)}, b' = \frac{b}{\mathrm{mdc}(a,b)}, c' = \frac{c}{\mathrm{mdc}(a,b)}$.
Existem $s, t \in \mathbb{Z}$ tais que $1 = sa' + tb'$ e $c' = c'sa' + c'tb'$.

$$\begin{aligned}
&\Rightarrow \quad a'(x - c's) + b'(y - c't) = c' - c' = 0 \\
&\Leftrightarrow \quad a'(x - c's) = b'(c't - y) \\
&\Rightarrow \quad a'|(c't - y) \qquad \text{porque } \mathrm{mdc}(a', b') = 1 \\
&\Rightarrow \quad c't - y = a'n, \qquad n \in \mathbb{Z} \\
&\Rightarrow \quad y = c't - a'n, \qquad n \in \mathbb{Z} \\
&\Rightarrow \quad x = c't + b'n, \qquad n \in \mathbb{Z}
\end{aligned}$$

A classe dos inteiros módulares

```
class modular
{
unsigned int modulo;
int numero;

int mdc(int, int);

public:
modular(unsigned int, int);
modular operator + (modular);
modular operator - (modular);
modular operator * (modular);
bool invertivel();
modulear inverso ();
}
```

A classe dos inteiros módulares

```
modular(unsigned int n, int a)
{
base=n;
numero=a%n;
};

modular modular::operator + (modular b)
{
if (b.base == base)
return modular(base,b.numero+numero)
};

bool invertivel()
{
return (mdc(numero,base)==1);
};
```

Algoritmo de Euclid em C++

```
int mdc(int a, int b, int* x, int* y)
{
int r0,s0,t0,r1,s1,t1;
r0=a; s0=1; t0=0;
r1=b; s1=0; t1=1;

while(r1!=0)
{
  int q = r0/r1;
  int h = r0%r1; r0=r1; r1=h;
  h=s0-q*s1; s0=s1; s1=h;
  h=t0-q*t1; t0=t1; t1=h;
};
(*x)=s0;
(*y)=t0;
return r0;
};

int main()
{
int a,b,x,y;
x=0;
y=0;
cin >> a >> b;
cout << "mdc=" << mdc(a,b,&x,&y);
cout << " = " << x << "*" << a << " + "<<  y << "*" << b << endl;
}
```

2.5 O Algoritmo de Karatsuba

Multiplicação rápida de polinómios.

Anatolii Karatsuba (1937-2008)

2.5 O Algoritmo de Karatsuba

Multiplicações são "caras". O algoritmo classico da mulitplicação de dois polinómios de grau n precisa $O(n^2)$ multiplicações:

$$\left(\sum_{i=0}^{n} a_i x^i\right)\left(\sum_{j=0}^{n} b_j x^j\right) = \sum_{i=0}^{n}\sum_{j=0}^{n} a_i b_j x^{i+j}$$

Podemos fazer melhor ?

2.5 O Algoritmo de Karatsuba

Se $f = f_1 x^m + f_0$ e $g = g_1 x^m + g_0$ com f_0, f_1, g_0, g_1 polinómios de grau menor que m.

$$fg = f_1 g_1 x^{2m} + (f_1 g_0 + f_0 g_1) x^m + f_0 g_0 \qquad \text{4 multiplicações}$$

2.5 O Algoritmo de Karatsuba

Se $f = f_1 x^m + f_0$ e $g = g_1 x^m + g_0$ com f_0, f_1, g_0, g_1 polinómios de grau menor que m.

$$\begin{aligned} fg &= f_1 g_1 x^{2m} + (f_1 g_0 + f_0 g_1) x^m + f_0 g_0 \qquad \textcolor{red}{\text{4 multiplicações}} \\ &= \underbrace{f_1 g_1}_{u} x^{2m} + (\underbrace{(f_0 + f_1)(g_0 + g_1)}_{w} - \underbrace{f_1 g_1}_{u} - \underbrace{f_0 g_0}_{v}) x^m + \underbrace{f_0 g_0}_{v} \end{aligned}$$

2.5 O Algoritmo de Karatsuba

Se $f = f_1 x^m + f_0$ e $g = g_1 x^m + g_0$ com f_0, f_1, g_0, g_1 polinómios de grau menor que m.

$$\begin{aligned} fg &= f_1 g_1 x^{2m} + (f_1 g_0 + f_0 g_1) x^m + f_0 g_0 && \text{4 multiplicações} \\ &= \underbrace{f_1 g_1}_{u} x^{2m} + (\underbrace{(f_0 + f_1)(g_0 + g_1)}_{w} - \underbrace{f_1 g_1}_{u} - \underbrace{f_0 g_0}_{v}) x^m + \underbrace{f_0 g_0}_{v} \\ &= u x^{2m} + (w - u - v) x^m + v && \textbf{3 multiplicações} \end{aligned}$$

onde

$$u = f_1 g_1 \qquad v = f_0 g_0 \qquad w = (f_0 + f_1)(g_0 + g_1)$$

2.5 O Algoritmo de Karatsuba

Se $f = f_1 x^m + f_0$ e $g = g_1 x^m + g_0$ com f_0, f_1, g_0, g_1 polinómios de grau menor que m.

$$\begin{aligned} fg &= f_1 g_1 x^{2m} + (f_1 g_0 + f_0 g_1) x^m + f_0 g_0 \qquad \text{4 multiplicações} \\ &= \underbrace{f_1 g_1}_{u} x^{2m} + \big(\underbrace{(f_0 + f_1)(g_0 + g_1)}_{w} - \underbrace{f_1 g_1}_{u} - \underbrace{f_0 g_0}_{v}\big) x^m + \underbrace{f_0 g_0}_{v} \\ &= u x^{2m} + (w - u - v) x^m + v \qquad \textbf{3 multiplicações} \end{aligned}$$

onde

$$u = f_1 g_1 \qquad v = f_0 g_0 \qquad w = (f_0 + f_1)(g_0 + g_1)$$

pois

$$w = (f_0 + f_1)(g_0 + g_1) = f_0 g_0 + f_1 g_0 + f_0 g_1 + f_1 g_1 = u + v + (f_0 g_1 + f_1 g_0).$$

2.5 O Algoritmo de Karatsuba

Dividir e conquistar

Input: polinómios f, g

Output: $Karatsuba(f, g) = fg$

$m \leftarrow \lceil max(grau(f), grau(g))/2 \rceil$

if $m < 2$ **then**

  return $f \cdot g$ multiplicação usual

else

  $u \leftarrow Karatsuba(f_1, g_1)$

  $v \leftarrow Karatsuba(f_0, g_0)$

  $w \leftarrow Karatsuba(f_0 + f_1, g_0 + g_1)$

  return $u \cdot x^{2m} + (w - u - v) \cdot x^m + v$

end if

2.5 O Algoritmo de Karatsuba

Dividir e conquistar

Teorema

O algoritmo de Karatsuba precisa $O(n^{\log_2(3)})$ multiplicações.

2.5 O Algoritmo de Karatsuba

Dividir e conquistar

Teorema

O algoritmo de Karatsuba precisa $O(n^{\log_2(3)})$ multiplicações.

$n = 2^k$	Clássico 4^k	Karatsuba 3^k	percentagem
4	16	9	$\sim 56\%$
8	64	27	$\sim 42\%$
16	256	81	$\sim 31\%$
32	1024	243	$\sim 23\%$
64	4096	729	$\sim 17\%$
128	16384	2187	$\sim 13\%$
256	65536	6561	$\sim 10\%$

Implementação do algoritmo de Karatsuba

Decomposição $f = f_1 x^m + f_0$ pelo quociente e resto:

$$f_1 = f / x^m \qquad f_0 = f \% x^m$$

que só "cortam" o vector dos coeficientes em duas partes, i.e. se $f = (a_0, a_1, \ldots, a_n)$ então

$$f_0 = f \% x^m = (a_0, \ldots, a_{m-1})$$

$$f_1 = f / x^m = (a_m, \ldots, a_n).$$

A multiplicação por x^k é um "shift", i.e.

$$f \cdot x^k = (\underbrace{0, \ldots, 0}_{k-\text{vezes}}, a_0, \ldots, a_n).$$

```
polinomio polinomio::karatsuba(polinomio f, polinomio g)
{
  int maximo=max(f.coef.size(),g.coef.size());
  int d=maximo/2 + maximo%2;

  if (d<2) return f*g;
  else {
    polinomio f0=f.rem(d);     polinomio f1=f.quo(d);
    polinomio g0=g.rem(d);     polinomio g1=g.quo(d);

    polinomio u=karatsuba(f1,g1);
    polinomio v=karatsuba(f0,g0);
    polinomio w=karatsuba(f0+f1,g0+g1);

    return u.shift(2*d)+(w-u-v).shift(d)+v;
  };
};
```

2.5 O Algoritmo de Karatsuba

para inteiros

Input: inteiros $a = (a_0, \dots, a_n), b = (b_0, \dots, b_k)$ na base $\mathbf{q}$

Output: $Karatsuba(a, b) = ab$

$m \leftarrow \lceil max(n, k)/2 \rceil$

if $m < 2$ **then**

  return $a \cdot b$ multiplicação usual

else

  $a' \leftarrow (a_0, \dots, a_{m-1}), \qquad a'' \leftarrow (a_m, \dots, a_n);$

  $b' \leftarrow (b_0, \dots, b_{m-1}), \qquad b'' \leftarrow (b_m, \dots, b_k);$

  $u \leftarrow Karatsuba(a', b')$

  $v \leftarrow Karatsuba(a'', b'')$

  $w \leftarrow Karatsuba(a' + a'', b' + b'')$

  return $v \cdot \mathbf{q}^{2m} + (w - u - v) \cdot \mathbf{q}^m + u$

end if