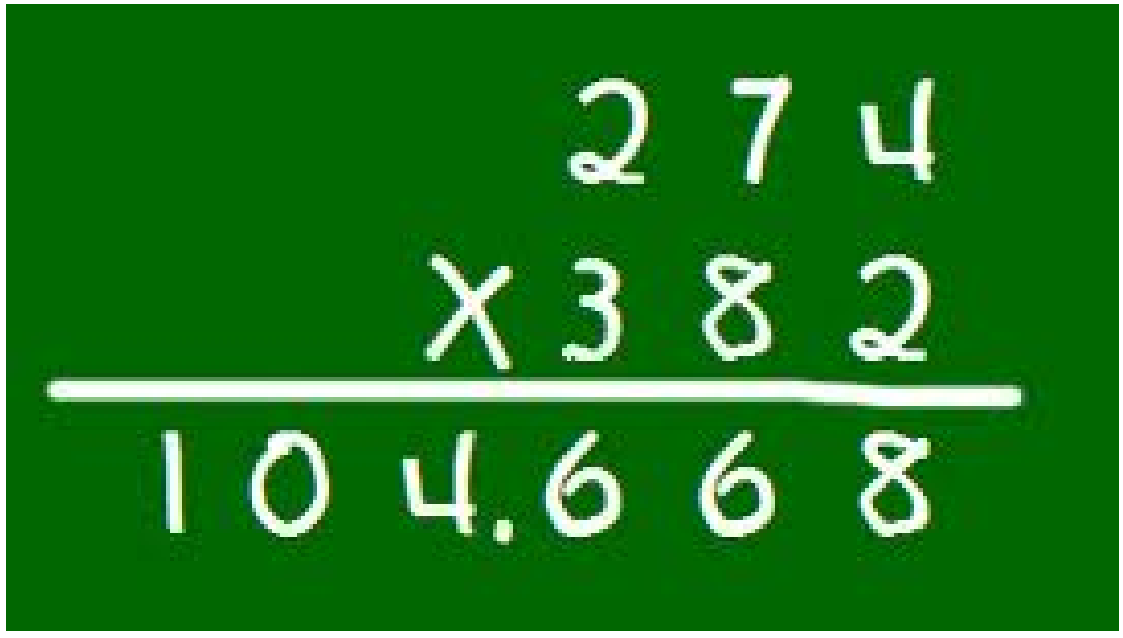
Multiplicação de inteiros
Algoritmo de Karatsuba

Paulo Feofiloff

Instituto de Matemática e Estatística
Universidade de São Paulo

25/7/2011

Universidade Federal do ABC



$$35871227428009 \times 11234908764388 \ = \ ?$$

Problema

Dados números inteiros positivos u e v com n dígitos cada
calcular o produto $u \times v$.

- exemplo: 99998888×77776666
- exemplo: 99998888×00076666
- imagine-se fazendo as contas com lápis e papel, dígito a dígito
- quanto tempo vai levar?
- tempo $\cong$ número de operações elementares
- *operação elementar* $=$ adição e multiplicação de dígitos
- n grande: aplicações à criptografia, fast Fourier transform, etc.

Se u e v têm n dígitos cada então

- $u \times v$ tem no máximo $2n$ dígitos
- $u + v$ tem no máximo $n + 1$ dígitos

Exemplos:

- $9999 \times 9999 = 9998\,0001$
- $9999 + 9999 = 1\,9998$

Algoritmo de multiplicação ordinário

Adição ordinária

$$\begin{array}{rl} 9\,9\,9\,9 & u \\ 7\,7\,7\,7 & v \\ \hline 1\,7\,7\,7\,6 & u+v \end{array}$$

- base do algoritmo: adição de dígitos
- n operações elementares

Multiplicação ordinária

$$
\begin{array}{rl}
9\,9\,9\,9 & u \\
7\,7\,7\,7 & v \\
\hline
6\,9\,9\,9\,3 & \\
6\,9\,9\,9\,3 & \\
6\,9\,9\,9\,3 & \\
6\,9\,9\,9\,3 & \\
\hline
7\,7\,7\,6\,2\,2\,2\,3 & u \times v
\end{array}
$$

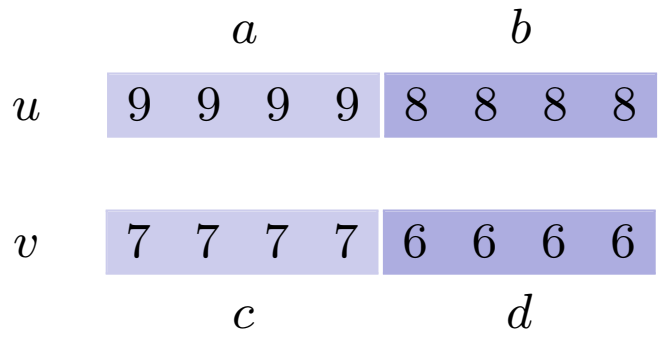
- base do algoritmo: multiplicação e adição de dígitos
- n^2 operações elementares (na verdade, $2n^2 + n$)
- n vezes mais lenta que adição
- n^2 é lento: $(2n)^2 = 4\,n^2$ e $(10n)^2 = 100\,n^2$

Desafio:

- inventar um algoritmo de multiplicação mais rápido
- há 50 anos acreditava-se que não existe algoritmo mais rápido
- acreditava-se que n^2 operações elementares são inevitáveis
- vou mostrar que $n^{1.6}$ operações elementares são suficientes

Divisão e conquista

"É mesmo? O *meu* povo sempre diz *multiplique* e conquiste."



- $u = a \cdot 10^{n/2} + b$
- $v = c \cdot 10^{n/2} + d$
- $u \times v = (a \times c) \cdot 10^n + (a \times d + b \times c) \cdot 10^{n/2} + b \times d$
- estamos supondo n par

$$
\begin{array}{rl}
99998888 & u \\
77776666 & v \\
 & \\
9999 & a \\
\quad 8888 & b \\
7777 & c \\
\quad 6666 & d \\
 & \\
77762223 & ac \\
\quad 135775310 & ad + bc \\
\quad 59247408 & bd \\
 & \\
7777580112347408 & x
\end{array}
$$

bom algoritmo para calculadora de bolso!

Número de operações elementares:

- $u \times v \;=\; (a \times c) \cdot 10^n \;+\; (a \times d + b \times c) \cdot 10^{n/2} \;+\; b \times d$
- 4 multiplicações de tamanho $n/2$ (mais 3 adições)
- operações elementares: $4\,(n/2)^2 = n^2$
- não ganhamos nada...

Tente, então, repetir o truque recursivamente

Esqueleto do algoritmo:

DIVIDA-E-CONQUISTE $(u,\ v,\ n)$

```
1  se n = 1
2      então devolva u × v
3      senão m ← n/2
4            a ← ⌊u/10^m⌋
5            b ← u mod 10^m
6            ...
8            ...
13           devolva x
```

Implementação: não leve as expressões $\lfloor u/10^m \rfloor$ e $u \bmod 10^m$ ao pé da letra

Divisão e conquista (n é potência de 2)

```
DIVIDA-E-CONQUISTE (u, v, n)
1  se n = 1
2      então devolva u × v
3      senão m ← n/2
4            a ← ⌊u/10^m⌋
5            b ← u mod 10^m
6            c ← ⌊v/10^m⌋
7            d ← v mod 10^m
8            ac ← DIVIDA-E-CONQUISTE (a, c, m)
9            bd ← DIVIDA-E-CONQUISTE (b, d, m)
10           ad ← DIVIDA-E-CONQUISTE (a, d, m)
11           bc ← DIVIDA-E-CONQUISTE (b, c, m)
12           x ← ac · 10^(2m) + (ad + bc) · 10^m + bd
13           devolva x
```

Número $T(n)$ de operações elementares:

- $T(1) = 1$ e $T(n) = 4\,T(n/2) + n$ para $n > 1$
- queremos uma "fórmula"
- solução da recorrência:

$$\begin{aligned}
T(n) &= 4\,T(n/2) + n \\
&= 4\left(4\,T(n/4) + n/2\right) + n \\
&= 16\,T(n/4) + 3n \\
&= 16\left(4\,T(n/8) + n/4\right) + 3n \\
&= 64\,T(n/8) + 7n \\
&= (2^j)^2\,T(n/2^j) + (2^j - 1)n \\
&= n^2\,T(n/n) + (n-1)n \\
&= 2n^2 - n
\end{aligned}$$

- $T(n) = 2n^2 - n$
- $T(n)$ é proporcional a n^2
- tão lento quanto o algoritmo ordinário...

Truque de Karatsuba

Divisão e conquista:

- $u = a \cdot 10^{n/2} + b \qquad v = c \cdot 10^{n/2} + d$
- $u \times v = (a \times c) \cdot 10^{n} + (a \times d + b \times c) \cdot 10^{n/2} + b \times d$
- 4 multiplicações de tamanho $n/2$ (e 3 adições)

Truque

- $y = (a + b) \times (c + d) = a \times c + a \times d + b \times c + b \times d$
- $u \times v = (a \times c) \cdot 10^{n} + (y - a \times c - b \times d) \cdot 10^{n/2} + b \times d$
- 3 multiplicações de tamanho $n/2$ (e 6 adições)

Estamos supondo n par

99998888	u
77776666	v
77762223	ac
59247408	bd
18887	$a+b$
14443	$c+d$
272784941	y
135775310	$y-ac-bd$
7777580112347408	x

Número de operações elementares:

- 3 multiplicações de tamanho $n/2$
- $3\,(n/2)^2 = 0.75\,n^2$ operações elementares
- ganhamos 25% com *uma* aplicação do truque
- para ganhar mais, repita o truque recursivamente

Algoritmo de Karatsuba (rascunho)

RASCUNHO-DE-KARATSUBA $(u,\ v,\ n)$

1 se $n = 1$
2 então devolva $u \times v$
3 senão $m \leftarrow n/2$
5 $a \leftarrow \lfloor u/10^m \rfloor$
6 $b \leftarrow u \bmod 10^m$
7 $c \leftarrow \lfloor v/10^m \rfloor$
8 $d \leftarrow v \bmod 10^m$
9 $ac \leftarrow$ RASCUNHO-DE-KARATSUBA $(a,\ c,\ m)$
10 $bd \leftarrow$ RASCUNHO-DE-KARATSUBA $(b,\ d,\ m)$
11 $y \leftarrow$ RASCUNHO-DE-KARATSUBA $(a+b,\ c+d,\ m+1)$
12 $x \leftarrow ac \cdot 10^{2m} + (y - ac - bd) \cdot 10^m + bd$
13 devolva x

Idéia básica correta, mas tem erros técnicos

Número de operações elementares:

- $T(n) = 3\,T(n/2) \;+\; n$ para $n = 2, 4, 8, 16, \ldots$ e $T(1) = 1$
- solução da recorrência: $T(n) = 3\,n^{\lg 3} - 2n$
- prova:
$$\begin{aligned} T(n) &= 3\,T(n/2) \;+\; n \\ &= 3\,(3\,(n/2)^{\lg 3} - 2(n/2)) \;+\; n \\ &= 3\,(n^{\lg 3} - n) \;+\; n \\ &= 3\,n^{\lg 3} \;-\; 2n \end{aligned}$$
- $T(2^j) = 3 \cdot 3^j - 2 \cdot 2^j$
- conclusão: $T(n)$ é proporcional a $n^{\lg 3}$

n	n^2	$n^{\lg 3}$	
8	64	27	42%
16	256	81	32%
32	1024	243	24%
64	4096	729	18%
128	16384	2187	13%
256	65536	6561	10%
512	262144	19683	8%
1024	1048576	59049	6%
2048	4194304	177147	4%
2^j	4^j	3^j	

- $\lg 3 \approx 1.584$
- $n^{\lg 3} \approx n\sqrt{n}$
- $n^{\lg 3}$ cresce mais devagar que n^2
- Karatsuba é mais rapido que o algoritmo ordinário (quando n é grande)

Algoritmo de Karatsuba: versão final, n arbitrário

```
KARATSUBA (u, v, n)
1  se n ≤ 3
2      então devolva u × v
3      senão m ← ⌈n/2⌉
4            a ← ⌊u/10^m⌋
5            b ← u mod 10^m
6            c ← ⌊v/10^m⌋
7            d ← v mod 10^m
8            ac ← KARATSUBA (a, c, m)
9            bd ← KARATSUBA (b, d, m)
10           y ← KARATSUBA (a + b, c + d, m + 1)
11           x ← ac · 10^{2m} + (y − ac − bd) · 10^m + bd
12           devolva x
```

Linha 10: $m + 1 < n$ pois $n > 3$

Testes de Liu, Huang, Lei:

	milissegundos		
n	ordinário	Karatsuba	
8	0.000	0.008	–
16	0.002	0.010	500%
32	0.007	0.019	271%
64	0.027	0.045	167%
128	0.101	0.128	127%
256	0.388	0.368	95%
512	1.541	1.111	72%
1024	6.079	3.400	56%
2048	24.460	12.000	49%

Quem é/foi Karatsuba?

- A.A. Karatsuba (1937–2008), matemático russo ▸ homepage
- Karatsuba descobriu o algoritmo quando tinha 23 anos
- o artigo foi publicado em 1962 sob os nomes de Karatsuba e Ofman

O algoritmo de Karatsuba é o mais rápido?

- existem algoritmos de multiplicação ainda mais rápidos
- algoritmo Toom-Cook: $n^{1.465}$
- algoritmo Schönhage-Strassen: $n \lg n \lg \lg n$

Conclusão

- nem sempre um algoritmo óbvio é o melhor
- algoritmos sofisticados podem ser mais rápidos (para n grande)
- matemática ajuda a projetar algoritmos mais sofisticados
- o poder do método de divisão-e-conquista
- o poder da recursão
- é útil saber resolver recorrências

Próximo passo: multiplicação rápida de matrizes

Referências

- Knuth, *The Art of Computer Programming*, 1997
- Brassard, and Bratley, *Algorithmics: Theory and Practice*, 1988
- Dasgupta, Papadimitriou, and Vazirani, *Algorithms*, 2006
- Kleinberg, and Tardos, *Algorithm Design*, 2005
- *Multiplication algorithm* ▸ Wikipedia
- *Karatsuba algorithm* ▸ Wikipedia

FIM

Obrigado pela atenção!

http://www.ime.usp.br/~pf/talks/UFABC-2011-07-25/
http://www.ime.usp.br/~pf/livrinho-AA/