On Karatsuba Multiplication Algorithm

Associate Prof. Fang

1. Introduction

Certain public key cryptographic algorithms such as RSA and ECC, the large integer multiplication is the basic operation of multiple precision integer arithmetic

1. Introduction

The literature about multiplication arithmetic covers:

- Classical Knuth multiplication($O(n^2)$)
- Karatsuba multiplication($O(n^{\log 3})$)
- Fast Fourier Transform trick($O(n\log n)$)
- Schönhage-Strassen trick($O(n\log n\log\log n)$)
- …

1. Introduction

Most of the multiplication techniques are “divide and conquer” tools.
But, Daniel J. Bernstein said:
“*It is a mistake to use a single method recursively all the way down to tiny problems. The optimal algorithm will generally use a different method for the next level of reduction, and so on.*"

1. Introduction

My short paper presents a new multiplication trick by using classical Knuth multiplication and Karatsuba multiplication, and finds the condition under which the efficiency of multiplication is optimal in theory and in practice.

2. Classical Knuth multiplication

Let $p=(u_1u_2\ldots u_n)_b$, $q=(v_1v_2\ldots v_m)_b$, the product is $w=pq=(w_1w_2\ldots w_{m+n})_b$. Here is the classical Knuth multiplication to compute the product w:

step1. $w_1, w_2, \ldots, w_{m+n} \leftarrow 0, j \leftarrow m;$

step2. $\text{if } v_j = 0 \text{ then } w_j \leftarrow 0 \text{ goto step6};$

step3. $i \leftarrow n, k \leftarrow 0;$

step4. $t \leftarrow u_i \times v_j + w_{i+j} + k,\ w_{i+j} \leftarrow t \bmod b,\ k \leftarrow \lfloor t/b \rfloor;$

step5. $i \leftarrow i-1, \text{ if } i>0 \text{ then goto step4 else } w_j \leftarrow k;$

step6. $j \leftarrow j-1, \text{ if } j>0 \text{ then goto step2 else exit};$

It is obvious that the time complexity of this algorithm is $O(mn)$.

3. Karatsuba multiplication

Let $p=(u_1u_2\ldots u_n)_b$, $q=(v_1v_2\ldots v_n)_b$. In 1963, Karatsuba wrote p×q as the following formula:

$$p\times q=r_1b^n+(r_2-r_1-r_0)b^{n/2}+r_0$$

where $r_0=p_0q_0$, $r_1=p_1q_1$, $r_2=(p_1+p_0)(q_1+q_0)$.

We can obtain the product by using "divide and conquer" method recursively. Let T(n) be computation time of multiplication p×q, we can get the recursion of time complexity easily:

$$T(n)=\begin{cases} 7, n=2 \\ 3T(n/2)+5n, n>2 \end{cases}$$

So we get $T(n)=9n^{\log 3}-10n=O(n^{\log 3})$

4. A new multiplication trick

Theorem 1. *There exists n such that the computational time of Knuth classical multiplication is less than that of Karatsuba multiplication.*

4. A new multiplication trick

Proof *Let T1(n) be computation time of classical Knuth multiplication and T2(n) be computation time of Karatsuba multiplication. According to the previous analysis, we have*

$T_1(n)=n^2,\ T_2(n)=9n^{\log 3}-10n$

There exists n such that $T_1(n) \le T_2(n)$, *that is*

$n^2 \le 9\, n^{\log 3} - 10n < 9 \cdot n^{\log 3}$

we can calculate $n < 2^{\frac{2\log 3}{2-\log 3}} \approx 2^{7.64} < 2^8 = 256$

Therefore, if n<256, then classical Knuth multiplication is more efficient than Karatsuba multiplication.

4. A new multiplication trick

Theorem 2: *the efficiency of Karatsuba multiplication is optimal when $n>16(n=2^k)$, Karatsuba multiplication algorithm is called recursively, and if $n=16$, then recursion call is returned, classical Knuth multiplication is used to compute the product of two smaller integers.*

Proof *Let $T(n)$ be computation time of Karatsuba multiplication. We assume that if $n>m$ then Karatsuba algorithm is called recursively, else classical Knuth multiplication is used. Therefore, we have*

$$T(n)=\begin{cases} m^2, n=m \\ 3T(n/2)+5n, n>m \end{cases}$$

4. A new multiplication trick

Let $n=2^k$, $h(k)=T(n)=T(2^k)$, $T(n)$ can be written as

$$h(k) = 3h(k-1) + 5 \bullet 2^k = 3(3h(k-2) + 5 \bullet 2^{k-1}) + 5 \bullet 2^k = \cdots$$

$$= 3^{k-i} h(i) + 5 \bullet 3^{k-(i-1)} \bullet 2^{i-1} + \cdots + 5 \bullet 3^0 \bullet 2^k$$

Let $m=2^i$, we get

$$h(k) = \frac{4^i + 10 \bullet 2^i}{3^i} \bullet n^{\log 3} - 10n$$

Let $f(i)=(4^i+10\cdot 2^i)/3^i$, the value of function $f(i)$ is minimum when

$$i = \left\lceil \frac{\log(10\log\frac{2}{3}) - \log\log\frac{4}{3}}{\log 2 - \log 3} \right\rceil = 4$$

That is, when $i=4$, $m=2^i=16$, the value of $T(n)$ is minimum.

$$T_{\min}(n) = \frac{416}{81} \bullet n^{\log 3} - 10n < T_2(n) = 9 \bullet n^{\log 3} - 10n$$

5. Experiment results and conclusion

Precondition: some simple assembly language codes may be called to compute the product of two 32-bit positive integers. The time complexity of this base operation is O(1).

```
_ _asm{
mov eax, x
xor edx, edx
mul  y
; Product in edx:eax
mov ebx, p
mov dword ptr [ebx], eax
mov dword ptr [ebx+4], edx
}
```

5. Experiment results and conclusion

Test environment:AMD Athlon CPU 1.1GHz, 256M RAM, Windows XP OS and MS Visual C++ 6.0 compiler.

5. Experiment results and conclusion

Table 1: the computation time comparison of three algorithms

Digits (radix 2^{32})	*Knuth*	*Karatsuba*	**New trick**
256	0.03	0.03	0.01
512	0.04	0.11	0.01
1024	0.17	0.381	0.05
2048	0.721	0.961	0.13
4096	2.734	2.874	0.381
8192	10.966	8.322	1.141

Where Digits is the length of multiplier integer in radix 2^{32} representation.

5. Experiment results and conclusion

Table 1 shows that the new multiplication trick obviously decreases computational time than that of the classical Knuth multiplication and Karatsuba multiplication.

References

1. R. L. Rivest, A. Shamir, L. Adleman, "*A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*". Communications of the ACM,1978,21(2), pp. 120-126.
2. Michael Rosing, *Implementing Elliptic Curve Cryptography*, Manning Publications Co. , Greenwich, 1999.
3. Anatoly A. Karatsuba, Y. Ofman, "*Multiplication of multi-digit numbers on automata*", Soviet Physics Doklady 7, 1963, pp. 595-596.
4. Dan Zuras, "*On Squaring and Multiplying Large Integers*", ARITH-11: IEEE Symposium on Computer Arithmetic, 1993, pp. 260-271. Reprinted as "*More on Multiplying and Squaring Large Integers*", IEEE Transactions on Computers, volume 43, number 8, August 1994, pp. 899-908.
5. E. Oran Brigham, *The fast Fourier transform and its applications*, Prentice-Hall, Englewood Cliffs, New Jersey, 1988.
6. A. SchÖnhage and V. Strassen, "*Schnelle Multiplikation großer Zahlen*", Computing 7, 1971, pp. 281-292.

References

7. **Daniel J. Bernstein. "*Multidigit Multiplication for Mathematicians*". http://cr.yp.to/papers/m3.pdf, 2001.08.11.**
8. **Donald E. Knuth, *The Art of Computer Programming, Vol 2 Seminumerical Algorithms (second edition)*, Addison-Wesley, Massachusetts, 1981.**
9. **Tom St Denis, *BigNum Math Implementing Cryptographic Multiple Precision Arithmetic*, SYNGRESS Publishing, 2003.**
10. **Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, *Introduction to Algorithms(Second Edition)*, The MIT Press, Massachusetts, 2001.**
11. **A. Menezes, P. van, Oorschot, S. Vanstone, *Handbook of Applied Cryptography*, CRC Press Inc., 1996.**

Thank you!