Asian Journal of Information Technology 5 (2) : 111-112, 2006
© Medwell Online, 2006

Complexity Analysis of Multiplication of Long Integers

M.Sadiq and Jawed Ahmed
Faculty of Engineering and Technology, Computer Engineering Section,
University Polytechnic Jamia Millia Islamia, New Delhi-110025, India

Abstract: Over the years computer scientists have identified a number of general techniques that often yield effective algorithms to solve large classes of problems. This paper presents divide and conquer algorithm based problem for the multiplication of long integers. In this paper, we have introduced a new technique to calculate the complexity for the multiplication of long integers.

Key words: Divide and conquer, multiplying long integers, complexity

INTRODUCTION

This is an expansion of the paper on squaring and multiplying large integers[1], presented at the IEEE symposium on computer arithmetic (ARITH-11) in July 1993.Perhaps the most important and most widely applicable technique for designing effective algorithm is a strategy called "divide and conquer".It consists of breaking a problem of size n into smaller problems in such a way that from solution to the smaller problems we can easily construct a solution to the entire problem.

The problem of multiplying long integers: In this study we have considered the problem of multiplying two n-bit integers X and Y. We know that the algorithm for multiplication of n bit integers usually taught in elementary school involves computing n partial products of size n and thus is an O (n^2) algorithm, if we count single bit or digit multiplications and additions as one step. One divide and conquer approach to integer multiplication would break each of X and Y into two integers of n/2 bits each as shown in (Fig.1)

$$X=A2^{n/2}+B$$
$$Y=C2^{n/2}+D$$

Fig. 1: Breaking n bit integers into n/2 bit pieces

The product of X and Y can now be written as

$$XY=AC2^{n}+(AD+BC)2^{n/2}+BD \quad (1)$$

If we want to compute XY in this straightforward way, we have to perform four multiplication of (n/2) bit integers (AC,AD,BC and BD), three additions of integers with at most 2n bits (corresponding to the three positive signs in (1)) and two shifts (multiplication by 2^n and $2^{n/2}$) .As these additions and shifts take O(n) steps, we can write the following recurrence for T(n), the total number of bit operations needed to multiply n bit integers according to (1).

$$T(1)=1$$
$$T(n)=4T(n/2)+cn \quad (2)$$

We can take the constant c in (2) to be1, so the driving function d(n) is just n and then deduce that the homogeneous and particular solution are both O(n^2). In (1) the asymptotic efficiency is thus no greater than for the elementary school method. But recall that for (2) we get an asymptotic improvement if we decrease the number of subproblems. It may be surprise that we can do so, but consider the following formula for multiplying X by Y.

$$XY=AC2^{n}+[(A-B)(D-C)+AC+BD]2^{n/2}+BD \quad (3)$$

Although (3) looks more complicated than (1) it requires only three multiplications of (n/2) bit integers, AC, BD and (A-B)(D-C), six additions or subtractions and two shifts. Since all but the multiplications take O(n) steps, the time T(n) to multiply n bit integers by (3)is given by

$$T(1)=1$$
$$T(n)=3T(n/2)+cn$$

Whose solution is $T(n)=O(n^{\log 3})=O(n^{1.59})$ and the base of log is 2[1].

A new approach to find out the complexity: The study on squaring and multiplying large integers by Dan Zuras presented at the IEEE symposium on computer arithmetic in July 1993, explain the 2 way, 3 way and 4 way method to calculate the complexity of the multiplication of large integers. The 2 way, 3 way and 4 way method means to split the number into 2, 3 and 4 parts, respectively.

If we split the number into two parts , so in that case the complexity is O (n^2)[2]. Now the question is, what would be the complexity of the multiplication of long integers if we divide the given long integer into 4, 5, 6, ----------k parts. With the help of the algorithm given in[1-3], we have designed the new technique to calculate the complexity for the multiplication of long integers. For this technique we can write the recurrence relation as

$$T(1) = 1$$
$$T(n) = p/q \quad (5)$$

Whose solution is T (n) = O $(n^{\log p})$ here the base of log isq Where p= (2*k-1) and q=k

After applying (5), we have got the results which are listed in Table 1.

RESULTS AND DISCUSSION

Whenever two long integers are multiplied using divide and conquer strategy and we want to find out its complexity, the given long integers are divided into number of parts (say k) and then (5) mentioned above is applied. In Table 1, the results have been summarized after splitting the long integers into 2 to 10 parts. We can conclude from Table 1 that higher the value of k parts lowers the complexity.

Table 1: The results have been summarized after splitting the long integers into 2 to 10 parts

S.N0.	k parts	Complexity
1	2	$O(\log^{1.59})$
2	3	$O(\log^{1.465})$
3	4	$O(\log^{1.404})$
4	5	$O(\log^{1.365})$
5	6	$O(\log^{1.338})$
6	7	$O(\log^{1.318})$
7	8	$O(\log^{1.302})$
8	9	$O(\log^{1.289})$
9	10	$O(\log^{1.279})$

REFERENCES

1. Zuras, D., 1993. On squaring and multiplying large integers, in 11th IEEE Symposium computer Arithmetic, pp: 260-271.
2. Aho, A.V., J.E. Hopcroft, J.D. Ullman, 2002. Data structures and algorithms, Pearson Education Singapore, Pvt Ltd. Delhi.
3. Zuras D., 1994. More on squaring and multiplying large integers, IEEE Transactions on computers, 43: 899-908.