Multiplying large integers – Karatsuba Algorithm

Let

$$A = (a_n a_{n-1} \dots a_1)_2 \quad \text{and} \quad B = (b_n b_{n-1} \dots b_1)_2$$

be binary representation of two integers A and B, $n = 2^k$ for some positive integer k. The aim here is to compute the binary representation of $A \cdot B$. Recall that the elementary school algorithm involves computing n partial products of $a_n a_{n-1} \dots a_1 a_0$ by b_i for $i = 1, \dots, n$, and so its complexity is in $O(n^2)$.

Naive algorithm

Question 1. Knowing the binary representations of A and B, devise a divide-and-conquer algorithm to multiply two integers.

Answer

A naive divide-and-conquer approach can work as follows. One breaks each of A and B into two integers of $n/2$ bits each:

$$A = \underbrace{\left(a_n \dots a_{n/2+1}\right)_2}_{A_1} \cdot 2^{n/2} + \underbrace{\left(a_{n/2} \dots a_1\right)_2}_{A_2}$$

$$B = \underbrace{\left(b_n \dots b_{n/2+1}\right)_2}_{B_1} \cdot 2^{n/2} + \underbrace{\left(b_{n/2} \dots b_1\right)_2}_{B_2}$$

The product of A and B can be written as

$$A \cdot B = A_1 \cdot B_1 \cdot 2^n + (A_1 \cdot B_2 + A_2 \cdot B_1) \cdot 2^{n/2} + A_2 \cdot B_2 \tag{1}$$

Question 2. Write the recurrence followed by the time complexity of the naive algorithm.

Answer

Designing a divide-and-conquer algorithm based on the equality (1) we see that the multiplication of two n-bit integers was reduced to

- four multiplications of $n/2$-bit integers ($A_1 \cdot B_1$, $A_1 \cdot B_2$, $A_2 \cdot B_1$, $A_2 \cdot B_2$)
- three additions of integers with at most $2n$ bits
- two shifts

Since these additions and shifts can be done in cn steps for some suitable constant c, the complexity of the algorithm is given by the following recurrence:

$$\begin{aligned} Time(1) &= 1 \\ Time(n) &= 4 \cdot Time\left(n/2\right) + cn \end{aligned} \tag{2}$$

Question 3. Deduce the asymptotic time complexity of the naive algorithm. Compare it to the classical school method.

> **Answer**
>
> Following the Master Theorem, the solution of (2) is $Time(n) = O\left(n^2\right)$. This is no improvement of the classical school method from the asymptotic point of view.

Karatsuba Algorithm

To get an improvement, one needs to decrease the number of subproblems, i.e., the number of multiplications of $n/2$-bit integers.

Question 4. Show that $(A_1 - A_2) \cdot (B_2 - B_1) + A_1B_1 + A_2B_2 = A_1B_2 + A_2B_1$. Design a new divide-and-conquer algorithm to multiply two integers.

> **Answer**
>
> Proving the equality is a straightforward calculus. The Karatsuba algorithm derives from the following formula
>
> $$A \cdot B = A_1B_1 \cdot 2^n + [A_1B_1 + A_2B_2 + (A_1 - A_2) \cdot (B_2 - B_1)] \cdot 2^{n/2} + A_2B_2 \tag{3}$$

Question 5. Give the asymptotic time complexity of the Karatsuba algorithm.

> **Answer**
>
> Although (3) looks more complicated than (1), it requires only
>
> - three multiplications of $n/2$-bit integers ($A_1 \cdot B_1$, $A_2 \cdot B_2$, $(A_1 - A_2) \cdot (B_2 - B_1)$)
> - four additions, and two subtractions of integers of at most $2n$ bits
> - two shifts
>
> Thus the divide-and-conquer algorithm based on (3) has the time complexity given by the recurrence
>
> $$\begin{aligned} Time(1) &= 1 \\ Time(n) &= 3 \cdot Time\left(n/2\right) + dn \end{aligned} \tag{4}$$
>
> for a suitable constant d. According to the Master Theorem the solution of (4) belongs to $O\left(n^{\log_2 3}\right)$ where $\log_2 3 \approx 1.59$. So the Karatsuba algorithm is asymptotically faster than the school method.