Faster Multiplication of Integers Using Decimal Notation

December 3, 2001

Let $A$ and $B$ be (positive, for simplicity) integers with $n = 2^k$ digits each (if one of the them has less than $2^k$ digits then prefix it with zeros to the left. For example, if $A = 24681357$ has eight digits and $B = 135792$ has six digits, write $B$ as 0013572.)

We saw in lecture that the “grade-school” algorithm in which we multiply $A$ by each digit of $B$ and add the intermediate results requires time $\sim n^2$ since we have to do $n^2$ multiplications of individual digits.

To get a faster algorithm, we first write

$$A = A_h 10^n/2 + A_\ell$$

and

$$B = B_h 10^n/2 + B_\ell,$$

where each of $A_h$, $A_\ell$, $B_h$, $B_\ell$ has at most $n/2 = 2^{k-1}$ digits. Then

$$A \cdot B = A_h \cdot B_h 10^n + (A_h \cdot B_\ell + A_\ell \cdot B_h) 10^{n/2} + A_\ell \ell \cdot B_\ell.$$

The above equation requires four multiplications of integers with $n/2$ digits, three additions of digits with a most $n + 2$ digits (which can be done in time $\sim n$) and one shift by $n$ digits and one shift by $n/2$ digits both of which can also be carried out in time $\sim n$.

This leads to a divide and conquer algorithm whose computing time $T(n)$ satisfies

$$T(n) = \begin{cases} c & \text{if } n = 1, \\
4T(n/2) + cn & \text{if } n > 1,
\end{cases}$$

for a constant $c > 0$.

By part (2) of the master theorem this leads to an algorithm that requires time $\sim n^2$, since $a = 4$ and $b = 2$ in the application of the master theorem and $\log_b a = \log 4 = 2$, which does not give an asymptotic improvement.

However, we would get an asymptotic improvement if we could reduce the number of multiplication to three and keep all the other needed operations (additions and shifts) linear since the master theorem tells us that such an algorithm would run in time $n^{\log 3} = n^{1.58\ldots}$. 
To get such an algorithm, we notice that
\[(A_h - A_\ell) \cdot (B_\ell - B_h) = A_h B_\ell + A_\ell B_h - A_h B_h - A_\ell B_\ell.\]

Thus, we can write,
\[A \cdot B = A_h B_h 10^n + [(A_h - A_\ell) \cdot (B_\ell - B_h) + A_h B_\ell + A_\ell B_h]10n/2 + A_\ell B_\ell,\]

which requires only the three multiplications \(A_h \cdot B_h, A_\ell \cdot B_\ell,\) and \((A_h - A_\ell) \cdot (B_\ell - B_h).\)

To be completed.