# 9 Karatsuba's Algorithm for Integer Multiplication

Given two $n$-bit integers $a$ and $b$ (in binary form), it takes $O(n)$ time to compute $a+b$ and $a-b$. However, the simple primary school method to compute $ab$ takes $\Theta(n^2)$ time. And our goal today was to improve this time complexity via divide-and-conquer. Naturally we choose the middle point $m = \lceil \frac{n}{2} \rceil$ to divide the two long integers $a$ and $b$ into $4$ shorter integers. We do this in a way so that

$$a = x \cdot 2^m + y, b = u \cdot 2^m + w,$$

where $x, y, u, w$ are $m$-bit integers. Now we have

$$ab = (x \cdot 2^m + y)(u \cdot 2^m + w) = xu \cdot 2^{2m} + (xw + yu) \cdot 2^m + yw.$$

As long as we can compute $xu$, $xw$, $yu$, and $yw$, we can get $ab$ with an additional $O(n)$ time. On the other hand, we can compute these 4 products via recursion. Therefore, we come up with our first divide-and-conquer algorithm.

DC-MULT$(n, a, b)$

IF $(n \leq 1)$ RETURN $ab$;

Let $m, x, y, u, w$ be defined as above;

$p =$ DC-MULT$(m, x, u)$; $q =$ DC-MULT$(m, y, w)$

$r =$ DC-MULT$(m, x, w)$; $s =$ DC-MULT$(m, y, u)$

RETURN $p \cdot 2^{2m} + (r + s) \cdot 2^m + q$

We see that besides 4 recursive calls, the procedure takes additional $O(n)$ time. Therefore, the time complexity $T(n)$ can be expressed by the following recurrence $T(n) = 4T(n/2) + \Theta(n)$ with boundary condition $T(n) = 1$ for $n \leq 1$. Solving this we get $T(n) = \Theta(n^2)$. It turns out we do not get any improvement from the simple primary school multiplication method.

Karatsuba, however, made a simple but clever observation in 1960, to reduce the number of recursive calls from 4 to 3. The observation is as follows.

Note that $p + q + r + s = (x + y)(u + w)$, which means $r + s = (x + y)(u + w) - p - q$. Therefore, if we compute $p$ and $q$ by 2 recursive calls, we only need 1 more recursive call to compute $(x+y)(u+w)$ and 2 more subtractions to get $r+s$. Since subtraction are relatively cheap, this saving is substantial. This only one small remark one need to make is that $(x + y)(u + w)$ is indeed a multiplication of two $(m+1)$-bit integers (rather than $m$-bit integers). However, this does not make much difference when solving the recurrence.

We describe the new divide-and-conquer algorithm as follows.

Karatsuba-MULT$(n, a, b)$

IF $(n < 5)$ RETURN $ab$;

Let $m, x, y, u, w$ be defined as above;

$p =$ Karatsuba-MULT$(m, x, u)$; $q =$ Karatsuba-MULT$(m, y, w)$

$t =$ Karatsuba-MULT$(m + 1, x + y, u + w)$

RETURN $p \cdot 2^{2m} + (t - p - q) \cdot 2^m + q$

Let $T(n)$ be the time complexity of the procedure above. We have the recurrence $T(n) = 3T(n/2)+ \Theta(n)$ with boundary condition $T(n) = O(1)$ for $n < 5$. Soving this and we get $T(n) = O(n^{\log_2 3}) = O(n^{1.585})$, which is asymptotically better than $n^2$.