Matrix Multiplication

We consider the problem of matrix multiplication. Assume for convenience that both are square matrices.
We all know how to compute the multiplication in $O(n^3)$ time for two $n \times n$ matrices.

A different cubic algorithm.

Assume $n = 2^k$.

$$A = \left[\begin{array}{c:c} A_{11} & A_{12} \\ \hdashline A_{21} & A_{22} \end{array}\right] \qquad B = \left[\begin{array}{c:c} B_{11} & B_{12} \\ \hdashline B_{21} & B_{22} \end{array}\right]$$

$$C = AB = \left[\begin{array}{c:c} C_{11} & C_{12} \\ \hdashline C_{21} & C_{22} \end{array}\right] = \left[\begin{array}{c:c} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\ \hdashline A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \end{array}\right]$$

So, to multiply two $n \times n$ matrices we multiply 2 $\frac{n}{2} \times \frac{n}{2}$ matrices 8 times. The overhead is 4 times additions of 2 $\frac{n}{2} \times \frac{n}{2}$ matrices.

$$T(n) = 8\,T\left(\frac{n}{2}\right) + O(n^2)$$

$$= O\left(n^{\log_2 8}\right) = O(n^3)$$

Observe that the time would remain the same even if the overhead increases to $O(n^{3-\epsilon})$ for $\epsilon > 0$.

Multiplying 2x2 matrices

The goal is to cut down ~~of~~ the number of multiplications from 8 to 7; and the # of additions/subtractions compensate for it.

Sequence of opr^n:

$p_1 := a_{11}(b_{12} - b_{22})$

$p_2 := (a_{11} + a_{12})\, b_{22}$

$p_3 := (a_{21} + a_{22})\, b_{11}$

$p_4 := a_{22}(b_{21} - b_{11})$

$p_5 := (a_{11} + a_{22})(b_{11} + b_{22})$

$p_6 := (a_{12} - a_{22})(b_{21} + b_{22})$

$p_7 := (a_{11} - a_{21})(b_{11} + b_{12})$

Checking correctness

$a_{11}b_{11} - a_{11}b_{22}$

$a_{11}b_{22} + a_{12}b_{22}$

$a_{21}b_{11} + a_{22}b_{11}$

$a_{22}b_{21} - a_{22}b_{11}$

$a_{11}b_{11} + a_{11}b_{22} + a_{22}b_{11} + a_{22}b_{22}$

$a_{12}b_{21} + a_{12}b_{22} - a_{22}b_{21} - a_{22}b_{22}$

$a_{11}b_{11} + a_{11}b_{12} - a_{21}b_{11} - a_{21}b_{12}$

$C_{11} := p_5 + p_4 - p_2 + p_6$

$C_{12} := p_1 + p_2$

$C_{21} := p_3 + p_4$

$C_{22} := p_5 + p_1 - p_3 - p_7$

$a_{11}b_{11} + a_{12}b_{21}$ ✓

$a_{11}b_{12} + a_{12}b_{22}$ ✓

$a_{21}b_{11} + a_{22}b_{21}$ ✓

$a_{21}b_{12} + a_{22}b_{22}$ ✓

We get the following recurrence

$$T(n) = 7\,T\left(\frac{n}{2}\right) + O(n^2)$$

$$= O(n^{\log_2 7}) = O(n^{2.808}).$$

Notes

(1) if n is not a power of 2 then all matrices A, B, C can be extended to size $m \times m$, for $m = 2^{\lceil \log_2 n \rceil}$.
The cost for multiplication does not increase for this padding

(2) The algorithm is due to Strassen, 1969.
In practice, the algorithm will probably beat the simple $O(n^3)$ algorithm for very large n. What is this large n?

(3) The complexity has since been improved to $O(n^{2.376})$.
The lower bound remains $\Omega(n^2)$.