Strassen's matrix multiplication

Thursday, June 16, 2022 9:51 AM

What is the complexity of multiplying two square matrices $A_{n\times n} * B_{n\times n}$?

Straightforward algo: $O(n^3)$.

Assume that $n = 2^k \cdot B$

$$\left[\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right]\left[\begin{bmatrix} e & f \\ g & h \end{bmatrix}\right]$$

$a, b, \ldots h$ are $\frac{n}{2}\times\frac{n}{2}$ square matrices

$$C = \begin{bmatrix} C_{11} = ae+bg & C_{12} = af+bh \\ C_{21} = ce+dg & C_{22} = cf+dh \end{bmatrix} = \begin{bmatrix} P_5+P_4-P_2+P_6 & P_1+P_2 \\ P_3+P_4 & P_1+P_5-P_3-P_7 \end{bmatrix}$$

8

ae → matrix multiplication: dimension is $n/2$.

How many scalar operations are needed for $ae+bg$? $\frac{n^2}{4}$ additions.

R.R. $T(n) = 8T\left(\frac{n}{2}\right) + 4 * \frac{n^2}{4} = 8T(n/2) + n^2$

Apply M.T. $a=8$ $b=2$ $f(n)=n^2$ $g(n) = n^{\log_2 8} = n^3$

Case 1 $g(n) >_p f(n)$. $\therefore T(n) = \Theta(n^3)$.

Examine: $T(n) = aT\left(\frac{n}{2}\right) + bn^2$

What determines the time complexity?

If I reduce a to that $a < 8$ the time complexity improves even if $b \gg 1$.

Strassen showed a way to accomplish $ae+bg$ etc. using fewer recursive calls.

$P_1 = a(f-h)$

$P_2 = (a+b)h$

$P_3 = (c+d)e$

$P_4 = d(g-e)$

$P_5 = (a+d)(e+h)$

$P_6 = (b-d)(g+h)$ → 10

$P_7 = (a-c)(e+f)$

P_3+P_4 $ce+de+dg-de$

The work done in addition to recursive calls: $18\left(\frac{n^2}{4}\right) = \frac{9}{2}n^2$

R.R.: $T(n) = 7T\left(\frac{n}{2}\right) + \frac{9}{2}n^2$

$a=7$ $b=2$ $f(n)=n^2$ $g(n) = \log_2 7$

$g(n) >_p f(n)$ Case 1) $T(n) = \Theta(n^{\log_2 7}) = n^{2.81}$

n^3 versus $(2n)^{2.81}$

18 submatrix $+s$ or $-s$

$18 \times \frac{n^2}{4} = \frac{9}{2}n^2$