Strassen’s Algorithm

Matrix Multiplication

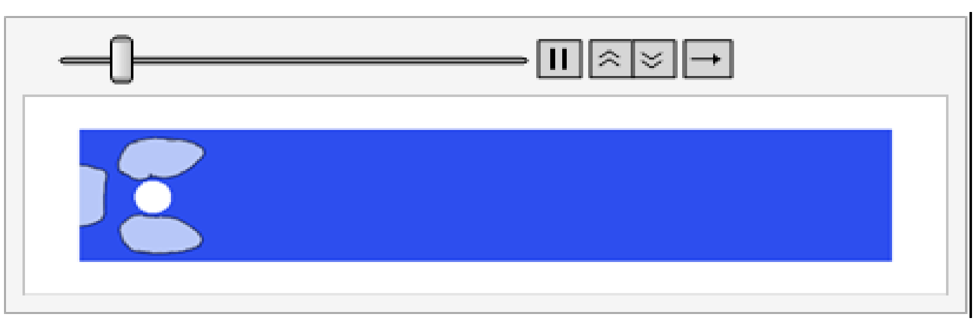
Problem: Given two matrices, A and B, determine AB.

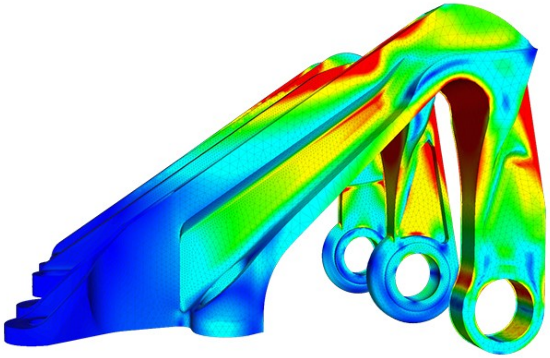
Example: $\begin{bmatrix} 4 & 0 & 0 \\ 0 & 5 & 2 \end{bmatrix} \begin{bmatrix} -1 \\ 3 \\ -2 \end{bmatrix} = \begin{bmatrix} -4 \\ 11 \end{bmatrix}$

Monkey: $\Theta(n^3)$

Lower bound: $\Omega(n^2)$

Applications: approximating solutions to differential equations/finite-element analysis





Linear Algebra Facts

<u>Math fact</u>: It is possible to split matrices into submatrices and then multiply the submatrices as if they were single entries.

$$\left[\begin{array}{c|cc} a & b & c \\ \hline d & e & f \end{array}\right] \left[\begin{array}{c} x \\ \hline y \\ z \end{array}\right] = \left[\begin{array}{c} ax + \left[\begin{array}{cc} b & c \end{array}\right] \left[\begin{array}{c} y \\ z \end{array}\right] \\ dx + \left[\begin{array}{cc} e & f \end{array}\right] \left[\begin{array}{c} y \\ z \end{array}\right] \end{array}\right]$$

$$= \left[\begin{array}{c} ax + by + cz \\ dx + ey + fz \end{array}\right]$$

<u>Note</u>: This clever math fact allows us to assume that we are multiplying square matrices of size $n \times n$ where n is a power of 2. If not, pad A and B by 0 entries.

$$\left[\begin{array}{c|c} \mathrm{A} & 0 \\ \hline 0 & 0 \end{array}\right] \left[\begin{array}{c|c} \mathrm{B} & 0 \\ \hline 0 & 0 \end{array}\right] = \left[\begin{array}{c|c} \mathrm{AB} & 0 \\ \hline 0 & 0 \end{array}\right]$$

A Clever Recursive Algorithm

Split A and B into 4 subsections each and multiply recursively.

$$A = \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right] \text{ and B=} \left[\begin{array}{c|c} B_{11} & B_{12} \\ \hline B_{21} & B_{22} \end{array}\right]$$

If A and B are both size $n \times n$, how big are the A_{ij} and B_{ij} matrices?

$\frac{n}{2} \times \frac{n}{2}$

$$AB = \left[\begin{array}{c|c} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\ \hline A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \end{array}\right]$$

Where does the recursion stop? $n = 1$

A Depressing Time Analysis

$$AB = \left[\begin{array}{c|c} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\ \hline A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \end{array}\right]$$

Let $T(n)$ be the number of single-register multiplications needed to multiply together two $n \times n$ matrices using our recursive algorithm.

$T(n) = 8T(\frac{n}{2})$ and $T(1) = 1$

$$T(n) = 8T(\tfrac{n}{2}) \Rightarrow T(\tfrac{n}{2}) = 8T(\tfrac{n}{4}) \Rightarrow T(\tfrac{n}{4}) = 8T(\tfrac{n}{8}) \ldots \Rightarrow$$

$$T(n) = 8T(\tfrac{n}{2}) = 8 \cdot 8T(\tfrac{n}{4}) = 8 \cdot 8 \cdot 8T(\tfrac{n}{8}) = \ldots$$

$$\ldots = 8 \cdot \ldots \cdot 8T(1) = 8 \cdot 8 \ldots \cdot 8$$

How many 8's are there? $\log_2 n \Rightarrow$

$T(n) = 8 \cdot \ldots \cdot 8 = 8^{\log_2 n} = n^{\log_2 8} = n^3$

Then Strassen Said...

In 1969, Strassen said:

$$M_1 = (A_{11} + A_{22})(B_{11} + B_{22})$$
$$M_2 = (A_{21} + A_{22})B_{11}$$
$$M_3 = A_{11}(B_{12} - B_{22})$$
$$M_4 = A_{22}(B_{21} - B_{11})$$
$$M_5 = (A_{11} + A_{12})B_{22}$$
$$M_6 = (A_{21} - A_{11})(B_{11} + B_{12})$$
$$M_7 = (A_{12} - A_{22})(B_{21} + B_{22})$$
$$\begin{bmatrix} M_1 + M_4 - M_5 + M_7 & M_3 + M_5 \\ M_2 + M_4 & M_1 + M_3 - M_2 + M_6 \end{bmatrix}$$

And the world said... What?

Then Strassen Said...

$$M_1 = (A_{11} + A_{22})(B_{11} + B_{22})$$
$$M_2 = (A_{21} + A_{22})B_{11}$$
$$M_3 = A_{11}(B_{12} - B_{22})$$
$$M_4 = A_{22}(B_{21} - B_{11})$$
$$M_5 = (A_{11} + A_{12})B_{22}$$
$$M_6 = (A_{21} - A_{11})(B_{11} + B_{12})$$
$$M_7 = (A_{12} - A_{22})(B_{21} + B_{22})$$

$$\begin{bmatrix} M_1 + M_4 - M_5 + M_7 & M_3 + M_5 \\ M_2 + M_4 & M_1 + M_3 - M_2 + M_6 \end{bmatrix}$$

$$M_3 + M_5 = A_{11}(B_{12} - B_{22}) + (A_{11} + A_{12})B_{22} =$$
$$A_{11}B_{12} + A_{12}B_{22}$$

Where have we seen that expression before?

Then Strassen Said...

$$M_1 = (A_{11} + A_{22})(B_{11} + B_{22})$$
$$M_2 = (A_{21} + A_{22})B_{11}$$
$$M_3 = A_{11}(B_{12} - B_{22})$$
$$M_4 = A_{22}(B_{21} - B_{11})$$
$$M_5 = (A_{11} + A_{12})B_{22}$$
$$M_6 = (A_{21} - A_{11})(B_{11} + B_{12})$$
$$M_7 = (A_{12} - A_{22})(B_{21} + B_{22})$$

$$\begin{bmatrix} M_1 + M_4 - M_5 + M_7 & M_3 + M_5 \\ M_2 + M_4 & M_1 + M_3 - M_2 + M_6 \end{bmatrix}$$

$$M_3 + M_5 = A_{11}(B_{12} - B_{22}) + (A_{11} + A_{12})B_{22} =$$
$$A_{11}B_{12} + A_{12}B_{22}$$

$$AB = \left[\begin{array}{c|c} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\ \hline A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \end{array}\right]$$

Then Strassen Said...

$$M_1 = (A_{11} + A_{22})(B_{11} + B_{22})$$
$$M_2 = (A_{21} + A_{22})B_{11}$$
$$M_3 = A_{11}(B_{12} - B_{22})$$
$$M_4 = A_{22}(B_{21} - B_{11})$$
$$M_5 = (A_{11} + A_{12})B_{22}$$
$$M_6 = (A_{21} - A_{11})(B_{11} + B_{12})$$
$$M_7 = (A_{12} - A_{22})(B_{21} + B_{22})$$

$$\begin{bmatrix} M_1 + M_4 - M_5 + M_7 & M_3 + M_5 \\ M_2 + M_4 & M_1 + M_3 - M_2 + M_6 \end{bmatrix}$$

$$AB = \left[\begin{array}{c|c} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\ \hline A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \end{array}\right]$$

This is a method for multiplying a 2×2 matrix in 7 multiplications instead of 8! So what?

A Surprisingly Happy Time Analysis

$$AB = \left[\begin{array}{c|c} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\ \hline A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \end{array}\right]$$

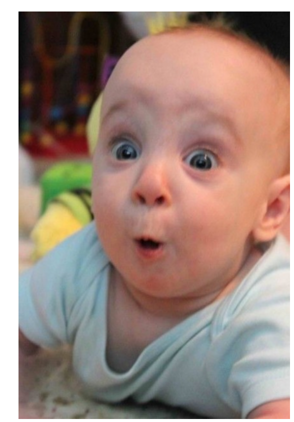
Let $T(n)$ be the number of single-register multiplications needed to multiply together two $n \times n$ matrices using our recursive algorithm.

$T(n) = 7T(\frac{n}{2})$ and $T(1) = 1$

$$T(n) = 7T(\tfrac{n}{2}) \Rightarrow T(\tfrac{n}{2}) = 7T(\tfrac{n}{4}) \Rightarrow T(\tfrac{n}{4}) = 7T(\tfrac{n}{8}) \ldots \Rightarrow$$

$$T(n) = 7T(\tfrac{n}{2}) = 7 \cdot 7T(\tfrac{n}{4}) = 7 \cdot 7 \cdot 7T(\tfrac{n}{8}) = \ldots$$

$$\ldots = 7 \cdot \ldots \cdot 7T(1) = 7 \cdot 7 \ldots \cdot 7$$



How many 7's are there? $\log_2 n \Rightarrow$

$$T(n) = 7 \cdot \ldots \cdot 7 = 7^{\log_2 n} = n^{\log_2 7} = n^{\sim 2.81}$$

The History of ω

Researchers in this area refer to the exponent of matrix multiplication as ω.

Strassen (1969): $\omega < 2.808$

Pan (1978): $\omega < 2.796$

Bini et al. (1979): $\omega < 2.78$

Schonhage (1981): $\omega < 2.548$

Schonhage, Pan (1981): $\omega < 2.522$

Romani (1982): $\omega < 2.517$

Coppersmith, Winograd (1981): $\omega < 2.49$

Strassen (1986): $\omega < 2.479$

Coppersmith, Winograd (1990): $\omega < 2.376$, presented conjecture which, if true, would imply that $\omega = 2$

Cohn, Umans et al. (2005): presented two conjectures which, if true, would imply that $\omega = 2$

Stothers (2010): $\omega < 2.374$

Alon, Shpilka, Umans (2011): showed that Coppersmith/Winograd conjecture and one of the Cohn/Umans conjectures presented above contradicts another conjecture by Erdos and Rado (sunflower conjecture) widely believed to be true; the other Cohn/Umans conjecture is still out there...

Virginia Vassilevska Williams (UC Berkeley, Stanford, 2011): $\omega < 2.3728642$

Francois Le Gall (2014): $\omega < 2.3728639$

Alman and Williams (2021): $\omega < 2.3728596$

(others in between and finally) Williams, Xu, Xu, and Zhou (2023): $\omega < 2.371552$

The Future of ω

Monkey: $\Theta(n^3)$

Strassen: $\Theta(n^{\sim 2.81})$

Vassilevska, etc.: $\Theta(n^{\sim 2.373})$

Lower bound: $\Omega(n^2)$

Most researchers in the this field believe that the "real" value of ω is... **2**!!! Take a minute to think about what that means...

In practice, most commercial large-scale mathematical packages, will use Strassen's algorithm if the matrices are large enough- dependent on the architecture of the computer running the computation- and then revert back to Monkey once the matrix sizes are small. In other words, the base case of Strassen's recursion is no longer 1 for these systems, but rather is redefined by the system to some threshold after which the Monkey algorithm kicks in.