Strassen’s Algorithm

Strassen’s Algorithm is for matrix multiplication.

Assume we are multiplying two n×n matrices:

$\mathrm{AB} = \mathrm{C}$

Start with n = 2:

$$C = \begin{pmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{pmatrix}$$

This takes 8 multiplications (and 4 additions).

Strassen’s insight was that we can do the $\mathrm{n} = 2$ case in only 7 multiplications (and 18 additions).

$$C = \begin{pmatrix} x_1 + x_4 - x_5 + x_7 & x_3 + x_5 \\ x_2 + x_4 & x_1 + x_3 - x_2 + x_6 \end{pmatrix}$$

Where:

$$\begin{array}{ll} x_1 = (a_{11} + a_{22}) * (b_{11} + b_{22}) & x_5 = (a_{11} + a_{12}) * b_{22} \\ x_2 = (a_{21} + a_{22}) * b_{11} & x_6 = (a_{21} - a_{11}) * (b_{11} + b_{12}) \\ x_3 = a_{11} * (b_{12} - b_{22}) & x_7 = (a_{12} - a_{22}) * (b_{21} + b_{22}) \\ x_4 = a_{22} * (b_{21} - b_{11}) & \end{array}$$

Note that commutativity of multiplication is not used. Hence we can generalise to matrices.

Suppose that $\mathrm{n} = 2^{\mathrm{k}}$. Divide up the matrices into four quadrants, each n/2×n/2:

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$$

Compute C_{ij} using the formulas for c_{ij}, and recursively compute each multiplication by further subdivision until you reach $n = 2$. We need to add/subtract 18 matrices of dimension n/2×n/2, and recursively perform 7 multiplications of n/2×n/2 matrices.

The number of arithmetic operations $A(k)$ for $n = 2^k$ is given by the recurrence relation:

$$A(0) = 1$$
$$A(k) = 7A(k-1) + 18(n/2)^2$$

We can solve this by repeated expansion and then summing the resulting geometric progression. The solution is:

$$A(k) = 7 \cdot 7^k - 6 \cdot 4^k = 7 \cdot 7^{\log n} - 6n^2 = 7 \cdot n^{\log 7} - 6n^2 \approx 7 \cdot n^{2.807} - 6n^2$$

So Strassen's matrix multiplication algorithm is $\Theta(n^{2.807})$

So far we have required $n = 2^k$. If $n \neq 2^k$, we can add an extra dummy row and column to keep the dimension even to allow subdivision.

Strassen's algorithm is an example of a **divide and conquer** algorithm:

- Divide problem into a subproblems of size n/b (here $a = 7$ and $b = 2$)
  - May take work to set up the subproblems (here matrix addition)
- Solve each subproblem recursively
- Then combine to get the result
  - Again, this may take work (here matrix addition)