Matrix Multiplication

X and Y are two $n \times n$ matrices

The running time of matrix multiplication algorithm is $O(n^3)$.

Can we do it better?

Can divide and Conquer help?

Matrix multiplication is particularly easy to break into sub-problems, because it can be performed block-wise. Curve X into four $(\frac{n}{2} \times \frac{n}{2})$ blocks and also Y as follows.

$$X = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

$$Y = \begin{bmatrix} E & F \\ G & H \end{bmatrix}$$

where A, B, C, D, E, F, G, H are the blocks

We can compute XY by considering blocks as single elements. that is,

$$XY = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} E & F \\ G & H \end{bmatrix} = \begin{bmatrix} AE + BG & AF + BH \\ CE + DG & CF + DH \end{bmatrix}$$

Now we can use divide and conquer strategy: to compute the size-n product XY, recursively compute eight size -n/2 products AE, BG, AF, BH, CE, DG, CF, DH and then do a few $O(n^2)$ additions.
Total running time:

$$T(n) = 8T(n/2) + O(n^2)$$

-------comes to $O(n^3)$.

For a quite while, this was widely believed to be the best running time possible.

Strassen Algorithm for matrix multiplication

In 1969, the German mathematician Volkar Strassen announced a significantly more efficient algorithm , based on divide-and-conquer. It was really a surprising fact!!!!

The key idea is to reduce the number of sub-problems using clever algebra.

It turns out that XY can be computed from just seven $n/2 \times n/2$ sub-problems via decomposition so tricky and intricate that one wonders how Strassen was ever able to discover it!!!!

$$XY = \begin{bmatrix} P_5 + P_4 - P_2 + P_6 & P_1 + P_2 \\ P_3 + P_4 & P_1 + P_5 - P_3 - P_7 \end{bmatrix}$$

where

$$\begin{gathered} P_1 = A(F - H) \\ P_2 = (A + B)H \\ P_3 = (C + D)E \\ P_4 = D(G - E) \\ P_5 = (A + D)(E + H) \\ P_6 = (B - D)(G + H) \\ P_7 = (A - C)(E + F) \end{gathered}$$

The new running time is:

$$T(n) = 7T(n/2) + O(n^2)$$

The master theorem works out to $O(n^{log_2 7}) \cong O(n^{2.81})$