Matrix Multiplication

$$\begin{bmatrix} n \times n \end{bmatrix} \cdot \begin{bmatrix} \circ \end{bmatrix} \rightarrow O(N^3)$$

하나가 n번 곱하고, n^2 개 있으므로 n^3 번 곱하기가 일어남.

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \cdot \begin{bmatrix} E & F \\ G & H \end{bmatrix}$$

Divide and Conquer 를 사용해도

호출에 8번 곱셈 필요 $\rightarrow 8x\ T(N/2) = O(2^{\log_8 N}) = O(N^3)$

Strassen - 행렬곱 빠르게

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

7가지 변수로 각각 곱셈을 한번씩만 사용해서 만든 후, 2 변수를 만으로

(7가지 변수들의 덧셈 뺄셈만 활용)

$= AE+BG$

$= AF+BH$ 들을 도출해낸다.

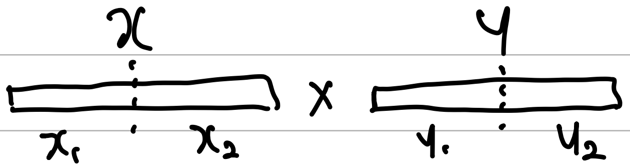
$= CE+DG$

$= CF+DH$

$$T(N) = 7T(N/2) = O(2^{\log_7 N}) = O(N^{2.8\cdots})$$

Karatsuba - 곱셈을 빠르게

n bits × n bits → $O(N^2)$



Divide and Conquer

$x \times y$

4번 곱셈 $4T(N/2) = O(2^{\log_4 N}) = O(N^2)$

$= (x_1 \times \underbrace{2^{\frac{n}{2}}}_{\text{자릿수}} + x_2) \times (y_1 \times 2^{\frac{n}{2}} + y_2) = x_1 y_1 2^n + (x_1 y_2 + x_2 y_1) 2^{\frac{n}{2}} + x_2 y_2$

그런데 여기서 한
곱셈 2번을 1번으로 줄일수 있다!

$(x_1 + x_2) \times (y_1 + y_2) = x_1 y_1 + x_1 y_2 + x_2 y_1 + x_2 y_2$

위에서 이미 구해짐

이 방법을 쓰면 한 층에서 곱셈 3번 사용

$3 \cdot T(N/2) = O(2^{\log_3 N}) = O(N^{\log_2 3}) = O(N^{1.\cdots})$