

Strassens algoritme

Matricer (repetition)

Matrix = firkant af tal:

$$\begin{bmatrix} 1 & 6 & 4 \\ 2 & 5 & 7 \\ 9 & 1 & 1 \end{bmatrix}$$

Ovenstående er en 3×3 matrix.

I dag: alle matricer er $n \times n$ kvadratiske matricer. (Dvs. n angiver sidelængden af matricerne.)

Matricer

Plus for matricer:

$$\begin{bmatrix} 1 & 6 & 4 \\ 2 & 5 & 7 \\ 9 & 1 & 1 \end{bmatrix} + \begin{bmatrix} 3 & 2 & 1 \\ 4 & 3 & 2 \\ 5 & 4 & 3 \end{bmatrix} = \begin{bmatrix} ? & ? & ? \\ ? & ? & ? \\ ? & ? & ? \end{bmatrix}$$

$$\begin{bmatrix} 1 & 6 & 4 \\ 2 & 5 & 7 \\ 9 & 1 & 1 \end{bmatrix} + \begin{bmatrix} 3 & 2 & 1 \\ 4 & 3 & 2 \\ 5 & 4 & 3 \end{bmatrix} = \begin{bmatrix} 1+3 & 6+2 & 4+1 \\ 2+4 & 5+3 & 7+2 \\ 9+5 & 1+4 & 1+3 \end{bmatrix} = \begin{bmatrix} 4 & 8 & 5 \\ 6 & 8 & 9 \\ 14 & 5 & 4 \end{bmatrix}$$

Tid? $\Theta(n^2)$.

Optimalt, da output er af størrelse n^2 .

Matricer

Gange for matricer:

$$\begin{bmatrix} 1 & 6 & 4 \\ 2 & 5 & 7 \\ 9 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 3 & 2 & 1 \\ 4 & 3 & 2 \\ 5 & 4 & 3 \end{bmatrix} = \begin{bmatrix} ? & ? & ? \\ ? & ? & ? \\ ? & ? & ? \end{bmatrix}$$

$$\begin{bmatrix} 1 & 6 & 4 \\ \underline{2} & \underline{5} & \underline{7} \\ 9 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 3 & 2 & \underline{1} \\ 4 & 3 & \underline{2} \\ 5 & 4 & \underline{3} \end{bmatrix} = \begin{bmatrix} ? & ? & ? \\ ? & ? & \textcolor{red}{33} \\ ? & ? & ? \end{bmatrix}$$

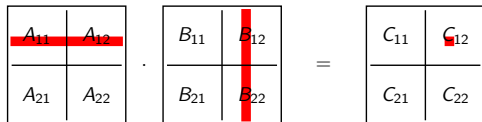
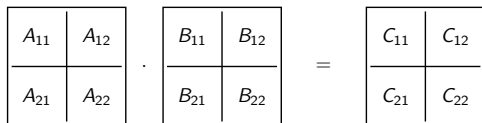
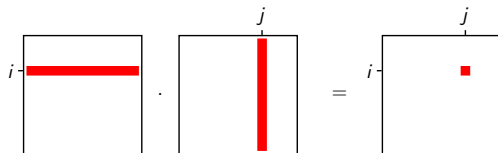
$$33 = 2 \cdot 1 + 5 \cdot 2 + 7 \cdot 3$$

$$\begin{bmatrix} 1 & 6 & 4 \\ 2 & 5 & 7 \\ \underline{9} & \underline{1} & \underline{1} \end{bmatrix} \cdot \begin{bmatrix} 3 & \underline{2} & 1 \\ 4 & \underline{3} & 2 \\ 5 & \underline{4} & 3 \end{bmatrix} = \begin{bmatrix} ? & ? & ? \\ ? & ? & 33 \\ ? & \textcolor{red}{25} & ? \end{bmatrix}$$

$$25 = 9 \cdot 2 + 1 \cdot 3 + 1 \cdot 4$$

Tid? $\Theta(n^3)$. Optimal?? Andre algoritmer??

Rekursiv algoritme for multiplikation?



Bemærk:

$$A_{11} \cdot B_{12} + A_{12} \cdot B_{22} = C_{12}$$

Rekursiv algoritme for multiplikation

$$\begin{array}{|c|c|} \hline A_{11} & A_{12} \\ \hline A_{21} & A_{22} \\ \hline \end{array} \cdot \begin{array}{|c|c|} \hline B_{11} & B_{12} \\ \hline B_{21} & B_{22} \\ \hline \end{array} = \begin{array}{|c|c|} \hline C_{11} & C_{12} \\ \hline C_{21} & C_{22} \\ \hline \end{array}$$

$$A_{11} \cdot B_{11} + A_{12} \cdot B_{21} = C_{11}$$

$$A_{11} \cdot B_{12} + A_{12} \cdot B_{22} = C_{12}$$

$$A_{21} \cdot B_{11} + A_{22} \cdot B_{21} = C_{21}$$

$$A_{21} \cdot B_{12} + A_{22} \cdot B_{22} = C_{22}$$

Additionerne: Lokalt arbejde på $n/2 \times n/2$ matricer. Tager $\Theta((n/2)^2) = \Theta(n^2)$ tid.

Multiplikationerne: Rekursive kald til matrixmultiplikation på $n/2 \times n/2$ matricer. (Base case: $n = 1 \Rightarrow$ multiplikation af tal.)

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

Rekursiv algoritme for multiplikation

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

Master theorem:

- ▶ $\alpha = \log_b(a) = \log_2(8) = 3$

- ▶ $f(n) = n^2$

$$n^2 = O(n^{\alpha-0.1}) \Rightarrow \text{Case 1}$$

$$T(n) = \Theta(n^\alpha) = \Theta(n^3)$$

Det samme som den almindelige algoritme. Øv.

Strassen [1969]

Beregn:

$$S_1 = B_{12} - B_{22}$$

$$S_6 = B_{11} + B_{22}$$

$$S_2 = A_{11} + A_{12}$$

$$S_7 = A_{12} - A_{22}$$

$$S_3 = A_{21} + A_{22}$$

$$S_8 = B_{21} + B_{22}$$

$$S_4 = B_{21} - B_{11}$$

$$S_9 = A_{11} - A_{21}$$

$$S_5 = A_{11} + A_{22}$$

$$S_{10} = B_{11} + B_{12}$$

Tid: $\Theta((n/2)^2) = \Theta(n^2)$, da både addition og subtraktion tager denne tid.

Strassen [1969]

Beregn:

$$P_1 = A_{11} \cdot S_1$$

$$P_2 = S_2 \cdot B_{22}$$

$$P_3 = S_3 \cdot B_{11}$$

$$P_4 = A_{22} \cdot S_4$$

$$P_5 = S_5 \cdot S_6$$

$$P_6 = S_7 \cdot S_8$$

$$P_7 = S_9 \cdot S_{10}$$

7 rekursive kald til matrixmultiplikation på $n/2 \times n/2$ matricer.

Strassen [1969]

Check nu at der gælder:

$$P_5 + P_4 - P_2 + P_6 = A_{11} \cdot B_{11} + A_{12} \cdot B_{21}$$

$$P_1 + P_2 = A_{11} \cdot B_{12} + A_{12} \cdot B_{22}$$

$$P_3 + P_4 = A_{21} \cdot B_{11} + A_{22} \cdot B_{21}$$

$$P_5 + P_1 - P_3 - P_7 = A_{21} \cdot B_{12} + A_{22} \cdot B_{22}$$

Eksempel på check:

$$P_1 + P_2 = A_{11} \cdot S_1 + S_2 \cdot B_{22} = A_{11} \cdot (B_{12} - B_{22}) + (A_{11} + A_{12}) \cdot B_{22} = A_{11} \cdot B_{12} + A_{12} \cdot B_{22}.$$

Dvs. output kan beregnes i $O(n^2)$ tid ud fra P_1, \dots, P_7 , eftersom

$$A_{11} \cdot B_{11} + A_{12} \cdot B_{21} = C_{11}$$

$$A_{11} \cdot B_{12} + A_{12} \cdot B_{22} = C_{12}$$

$$A_{21} \cdot B_{11} + A_{22} \cdot B_{21} = C_{21}$$

$$A_{21} \cdot B_{12} + A_{22} \cdot B_{22} = C_{22}$$

Strassen [1969]

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

Master theorem:

- ▶ $\alpha = \log_b(a) = \log_2(7) = 2.80735 \dots$
- ▶ $f(n) = n^2$

$$n^2 = O(n^{\alpha-0.1}) \Rightarrow \text{Case 1}$$

$$T(n) = \Theta(n^\alpha) = O(n^{2.81})$$

Bedre end den almindelige algoritme!