

Matrixmultiplikation

Algorithmus von STRASSEN

1. Juli 2025

Beim Algorithmus von Strassen für die Multiplikation von zwei $n \times n$ Matrizen lautet die Rekurrenz zur Ermittlung der Laufzeit $T(n)$ des Algorithmus

$$T(n) = 7 T\left(\frac{n}{2}\right) + c n^2.$$

Der Algorithmus halbiert in jedem rekursiven Aufruf die beiden $n \times n$ Matrizen zu vier $\frac{n}{2} \times \frac{n}{2}$ Matrizen. Substituieren wir n durch $\frac{n}{2}$, erhalten wir für

$$T\left(\frac{n}{2}\right) = 7 T\left(\frac{n}{4}\right) + c\left(\frac{n}{2}\right)^2 = 7 T\left(\frac{n}{4}\right) + \frac{c}{4} n^2.$$

Nach dem *ersten* rekursiven Aufruf erhalten wir mit $T\left(\frac{n}{2}\right)$ eingesetzt in $T(n)$ dann

$$\begin{aligned} T(n) &= 7 \left(7 T\left(\frac{n}{4}\right) + \frac{c}{4} n^2 \right) + c n^2 \\ &= 7^2 T\left(\frac{n}{4}\right) + \frac{7}{4} c n^2 + c n^2. \end{aligned}$$

Mit dem *zweiten* rekursiven Aufruf werden die vier $\frac{n}{2} \times \frac{n}{2}$ Matrizen wieder halbiert zu acht $\frac{n}{4} \times \frac{n}{4}$ Matrizen. Damit ist

$$T\left(\frac{n}{4}\right) = 7 T\left(\frac{n}{8}\right) + c\left(\frac{n}{4}\right)^2 = 7 T\left(\frac{n}{8}\right) + \frac{c}{16} n^2.$$

Wird $T\left(\frac{n}{4}\right)$ eingesetzt in $T(n)$ ergibt sich

$$\begin{aligned} T(n) &= 7^2 \left(7 T\left(\frac{n}{8}\right) + \frac{c}{16} n^2 \right) + \frac{7}{4} c n^2 + c n^2 \\ &= 7^3 T\left(\frac{n}{8}\right) + \frac{7^2}{4^2} c n^2 + \frac{7}{4} c n^2 + c n^2. \end{aligned}$$

Betrachten wir nun den k -ten rekursiven Aufruf finden wir für

$$T(n) = 7^k T\left(\frac{n}{2^k}\right) + c n^2 \sum_{i=0}^{k-1} \left(\frac{7}{4}\right)^i.$$

Zur Vereinfachung belassen wir es bei dem k -ten rekursiven Aufruf auch in $T(n)$ bei k und nicht $k+1$. Kleinere Matrizen als 1×1 Matrizen gibt es nicht, daher können die $n \times n$ Matrizen nur k mal halbiert werden. Der größte Wert, den k annehmen kann, ist $k = \log_2 n$. Damit ist

$$\begin{aligned} T(n) &= 7^{\log_2 n} T\left(\frac{n}{2^{\log_2 n}}\right) + c n^2 \sum_{i=0}^{\log_2 n - 1} \left(\frac{7}{4}\right)^i \\ &= n^{\log_2 7} T(1) + c n^2 \frac{\left(\frac{7}{4}\right)^{\log_2 n} - 1}{\frac{7}{4} - 1} \\ &= O\left(n^{2.8074}\right) + c n^2 \left(\frac{7}{4}\right)^{\log_2 n} \\ &= O\left(n^{2.8074}\right) + c n^2 \cdot n^{\log_2 \frac{7}{4}} \\ &= O\left(n^{2.8074}\right) + c n^{2.8074} \\ &= O\left(n^{2.8074}\right). \end{aligned}$$

Es folgt der Algorithmus 1 von Strassen als Pseudocode. Als Eingabe erhalten wir zwei $n \times n$ Matrizen. Der Einfachheit halber wird angenommen, dass n eine Zweierpotenz ist. Zu Beginn prüfen wir, ob die Größe der Matrizen bereits den Wert 1 hat. Haben die Matrizen den Wert 1, geben wir das Produkt AB zurück. In Zeile 4 und 5 werden die Matrizen A und B so

Algorithm 1 STRASSEN(A, B)

Eingabe: Matrizen A und B , beide $n \times n$, wobei $n = 2^k$, $k \in \mathbb{N}$

Ausgabe: Produktmatrix $C = AB$

```

1: if  $n = 1$  then  $C = AB$ 
2:   return  $C$ 
3: end if
4:  $A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$ 
5:  $B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}$ 
6:  $P_1 = \text{STRASSEN}(A_{11} + A_{22}, B_{11} + B_{22})$ 
7:  $P_2 = \text{STRASSEN}(A_{21} + A_{22}, B_{11})$ 
8:  $P_3 = \text{STRASSEN}(A_{11}, B_{12} - B_{22})$ 
9:  $P_4 = \text{STRASSEN}(A_{22}, B_{21} - B_{11})$ 
10:  $P_5 = \text{STRASSEN}(A_{11} + A_{12}, B_{22})$ 
11:  $P_6 = \text{STRASSEN}(A_{21} - A_{11}, B_{11} + B_{12})$ 
12:  $P_7 = \text{STRASSEN}(A_{12} - A_{22}, B_{21} + B_{22})$ 
13:  $C_{11} = P_1 + P_4 - P_5 + P_7$ 
14:  $C_{12} = P_3 + P_5$ 
15:  $C_{21} = P_2 + P_4$ 
16:  $C_{22} = P_1 - P_2 + P_3 + P_6$ 
17: return  $C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$ 

```

definiert, dass in den Zeilen 6 bis 12 die sieben Matrixmultiplikationen jeweils durchgeführt werden. In den Zeilen 13 bis 16 werden 4 Matrizen C_{ij} durch Addition und Subtraktion der Matrizen A_{ij} berechnet und in Zeile 17 Matrix C als Ergebnis zurückgegeben.

Als Idee zum Beweis der Korrektheit betrachten wir das Produkt $A \cdot B$ der zwei Matrizen A und B mit

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad \text{und} \quad B = \begin{pmatrix} e & f \\ g & h \end{pmatrix}.$$

Zur Vereinfachung beinhalten die Matrizen nur skalare Werte. Das Ergebnis $C = A \cdot B$ ist

$$C = \begin{pmatrix} ae + bg & af + bh \\ ce + dg & cf + dh \end{pmatrix}.$$

Der Algorithmus von Strassen berechnet P_i mit

$$\begin{aligned}
P_1 &= \text{STRASSEN}(a + d, e + h) &= ae + ah + de + dh \\
P_2 &= \text{STRASSEN}(c + d, e) &= ce + de \\
&\vdots \\
P_7 &= \text{STRASSEN}(b - d, g + h) &= bg + bh - dg - dh.
\end{aligned}$$

Dann werden C_{ij} berechnet mit

$$\begin{aligned}
C_{11} &= P_1 + P_4 - P_5 + P_7 &= ae + bg \\
C_{12} &= P_3 + P_5 &= af + bh \\
C_{21} &= P_2 + P_4 &= ce + dg \\
C_{22} &= P_1 - P_2 + P_3 + P_6 &= cf + dh,
\end{aligned}$$

wobei z.B. $C_{11} = (ae + ah + de + dh) + (dg - de) - (ah + bh) + (bg + bh - dg - dh) = ae + bg$.
Für einen formalen Beweis der Korrektheit verzichten wir auf die vollständige Induktion über $n \in \mathbb{N}$ der $n \times n$ Matrizen.