

## Max Wisniewski , Alexander Steen

Tutor: Lena Schlipf

**Aufgabe 1**

In dieser Aufgabe von wir zwei  $n \times n$  Matrizen schneller als mit der schulmethode Berechnen.

- (a) Erklären Sie in einem Satz, wie viele Additionen und Multiplikationen asymptotisch nötig sind, um zwei  $n \times n$  Matrizen mit dem Schulalgorithmus zu multiplizieren.

Bei der Schulmethode rechnet man einfach Zeilen mal Spalten, was bei  $n \cdot n$  Einträgen und pro Eintrag  $n$  Multiplikationen und  $n - 1$  Additionen sind.

$\Rightarrow n^3$  Multiplikationen und  $n^3 - n^2$  Additionen.

- (b) Zeigen Sie, warum die in der Übung gegebene Zerlegung von  $2 \times 2$  Matrizen auf beliebige Matrizen erweiterbar ist.

- (c) Beschreiben Sie die Laufzeit von Strassens Algorithmus mit einer Rekursionsgleichung und lösen Sie diese.

Für die Analyse nehmen wir das EKM.

Um zwei Matrizen zu Addieren (oder Subtrahieren) müssen wir die Komponenten addieren. Dafür brauchen wir also  $n^2$  Operationen.

Um nun die einzelnen Komponenten zu berechnen braucht man für  $P_1$  bis  $P_7$  10 dieser Additionen/Subtraktionen. Um nach der Rekursion auf  $r, s, t, u$  zu kommen benötigen wir noch einmal 9 Additionen/Subtraktionen. Diese operieren alle auf  $\frac{n}{2} \times \frac{n}{2}$  Matrizen.

$$\Rightarrow T(n) = \begin{cases} O(1) & , n = 2 \\ 7T(\frac{n}{2}) + 19n^2 & , \text{sonst} \end{cases}$$

Nun wenden wir darauf das Mastertheorem an:

Mit  $a = 7$ ,  $b = 2$ ,  $f(n) = 19n^2$

$$f(n) = 19n^2 \stackrel{\varepsilon=0.5}{=} O(n^{2.5-0.5}) = O(n^{\log_2 7}), \text{ da } \log_2 7 \approx 2.81$$

Damit können wir den ersten Fall des Mastertheorems anwenden:

$$\Rightarrow T(n) = \Theta(n^{\log_2 7}) \approx \Theta(n^{2.81})$$

**Aufgabe 2**

Geben Sie möglichst gute asymptotische Schranken für die folgenden rekursiv definierte Funktion  $T(n)$  an; dabei ist  $T(n) \in \Theta(1)$  für  $n \leq 2$ .

(a)  $T(n) = T(\frac{9}{10}n) + n$

Hier gilt  $a = 1$ ,  $b = \frac{10}{9}$ ,  $f(n) = n$

$$f(n) = n = \Omega(n^{0+\varepsilon}), \quad \text{mit } \varepsilon = 1,$$

bleibt zu zeigen, dass  $a \cdot f(\frac{n}{b}) \leq c \cdot f(n)$ :

$$1 \cdot f(\frac{9}{10}n) = \frac{9}{10}n \text{ gilt, mit } c = \frac{9}{10}$$

$$\Rightarrow T(n) = \Theta(n)$$

(b)  $T(n) = T(n-a) + T(a) + n$ ,  $a \leq 1$

Wie man schnell sieht, kann man hier das Mastertheorem nicht anwenden ( $n-a$  kann nicht auf  $\frac{n}{b}$  gebracht werden). Wir verlegen uns deshalb aufs einsetzen:

$$\begin{aligned} T(n) &= T(n-a) + T(a) + n \\ &= T(n-2a) + T(a) + n - a + T(a) + n \\ &= T(n-2a) + 2T(a) + 2n - a \\ &= T(n-3a) + 3T(a) + 3n - 3a \\ &= T(n-4a) + 4T(a) + 4n - a \cdot \sum_{i=1}^3 i \\ &\quad \text{Nach k Schritten} \\ &= T(n-ka) + kT(a) + kn - a \cdot \sum_{i=1}^{k-1} i \end{aligned}$$

Der Anker wird bei

$n - ka = 2 \Leftrightarrow k = \frac{n-2}{a}$  Damit löst sich die Gleichung auf zu:

$$T(n) = \Theta(1) + \frac{n-2}{a}T(a) + \frac{n-2}{a} \cdot n - a \cdot \frac{(\frac{n-2}{a}-1)(\frac{n-2}{a})}{2}$$

$T(a)$  ist konstant, da  $a \leq 1$  den Rekursionsanker trifft. Damit erhält man, wenn man alles ausklammert am Schluss:

$$T(n) = \Theta(n^2)$$

(c)  $T(n) = T(\sqrt{n}) + 1$

Wieder sieht man hier schnell das es nicht auf die Form des Mastertheorem kommen kann.

$$\begin{aligned} T(n) &= T(\sqrt{n}) + 1 \\ &= T(n^{\frac{1}{2}}) + 1 \\ &= T(n^{\frac{1}{4}}) + 2 \\ &\quad \text{Nach k Schritten} \\ &= T(n^{\frac{1}{2^k}}) + kn \end{aligned}$$

Den Anker erreichen wir dabei mit:  $n^{\frac{1}{2^k}} = 2 \Leftrightarrow 2^{2^k} = n \Leftrightarrow \log_2(\log_2 n) = k$   
Daraus ergibt sich eine Lauzeit von:

$$T(n) = \Theta(n \cdot \log \log n)$$

(d)  $T(n) = 2T(\frac{n}{4}) + \sqrt{n}$

$a = 2$ ,  $b = 4$ ,  $f(n) = \sqrt{n}$  Nun gilt:

$$\sqrt{n} = n^{\frac{1}{2}} = n^{\log_4 2} = \Theta(n^{\log_4 2})$$

Damit können wir das Mastertheorem anwenden.

$$\Rightarrow T(n) = \sqrt{n} \cdot \log n$$

(e)  $T(n) = 7T\left(\frac{n}{3}\right) + n^2$

Hier trifft leider keiner der Fälle des Mastertheorems zu.

$$\begin{aligned} T(n) &= 7T\left(\frac{n}{3}\right) + n^2 \\ &= 7^2T\left(\frac{n}{3^2}\right) + \frac{n^2}{(3^1)^2} + n^2 \\ &\quad \text{Nach } k \text{ Schritten} \\ &= 7^kT\left(\frac{n}{3^k}\right) + n^2 \cdot \sum_{i=0}^{k-1} \frac{1}{3^{2 \cdot i}} \end{aligned}$$

Den Anker erreichen wir bei:

$$\frac{n}{3^k} = 2 \Leftrightarrow n = 2 \cdot 3^k \Leftrightarrow k = \log_3 n - \log_3 2$$

Nach Einsetzen ergibt sich:

$$\begin{aligned} T(n) &= c \cdot 7^{\log_3 n - \log_3 2} + n^2 \cdot \sum_{i=0}^{\log_3 n - \log_3 2} \left(\frac{1}{3^2}\right)^i \\ &\stackrel{\text{summe konvergiert, } |q| < 1}{=} \frac{c}{7^{\log_3 2}} 7^{\log_3 n} + dn^2 \\ &= c' 7^{\log_3 n} + dn^2 \\ &= c' 7^{\frac{\log_7 n}{\log_7 3}} + dn^2 \\ &= c' n^{\frac{1}{\log_7 3}} + dn^2 = c' n^{\log_3 7} + dn^2 \end{aligned}$$

Nun ist  $\log_3 7 < 2$ . Deshalb können wir das ganze abschätzen.

$$\Rightarrow T(n) = \Theta(n^2)$$

(f)  $T(n) = 2T\left(\frac{n}{2}\right) + n \log n$

$$a = 2, b = 2, f(n) = n \log n$$

Nun gilt:

$$n^{\log_2 2} = n = O(n^{2-\varepsilon}), \varepsilon = 1$$

Damit können wir das Mastertheorem anwenden:

$$\Rightarrow T(n) = \Theta(n)$$

(g)  $T(n) = T(n-1) + \frac{1}{n}$

Wie man sieht, kann man diese Formel nicht auf Mastertheoremform bringen.

Deshalb setzen wir ein:

$$\begin{aligned} T(n) &= T(n-1) + \frac{1}{n} \\ &= T(n-2) + \frac{1}{n-1} + \frac{1}{n} \\ &\quad \text{Nach } k \text{ Schritten} \\ &= T(n-k) + \sum_{i=0}^{k-1} \frac{1}{n-i} \end{aligned}$$

Den Anker erreichen wir bei

$$n - k = 2 \Leftrightarrow k = n - 2$$

Also gilt:

$$\begin{aligned} T(n) &= \Theta(1) + \sum_{i=0}^{n-3} \frac{1}{n-i} \\ &= \Theta(1) - \frac{1}{1} - \frac{1}{2} + \sum_{i=0}^{n-1} \frac{1}{n-i} \\ &= \Theta(1) + \sum_{i=1}^n \frac{1}{i} \\ &= \Theta(1) + O(\log n) = O(\log n) \end{aligned}$$

Den letzten Umformungsschritt haben wir auf dem ersten Übungszettel gesehen. Das ganze würde sich noch genauer durch die Euler-Mascheroni-Konstante abschätzen, aber diese Näherung war uns nah genug.

### Aufgabe 3

- (a) Implementieren Sie Karatsubas Algorithmus, sowie die Schulmethode zur Multiplikation ganzer Zahlen. Vergleichen Sie die Laufzeit beider Algorithmen experimentell und schätzen Sie ab, ab welcher Eingabe Karatsuba besser ist.
- (b) Entwickeln Sie eine hybride Implementierung.