3. Aufgabenblatt zur Vorlesung

Höhere Algorithmik — WiSe 2011/12

Wolfgang Mulzer, Claudia Dieckmann, Romain Grunert, Tillmann Miltzow, Lena Schlipf

Abgabe am 11. November 2011 vor der Vorlesung in die Mitarbeiterschachteln

Aufgabe 1 Schnelle Matrizenmultiplikation *10 Punkte*

Die Multiplikation zweier $n \times n$ Matrizen ist eine fundamentale Operation, die nicht nur in numerischen Anwendungen ständig benötigt wird. In dieser Aufgabe werden wir eine Methode kennen lernen, die ähnlich zu Karatsubas Algorithmus die Schulmethode mit Hilfe einer magischen Rekursion verbessert.

(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.

(b) Volker Strassen hat einen Algorithmus zur schnelleren Matrizenmultiplikation entwickelt. Er funktioniert wie folgt (der Einfachheit halber nehmen wir an, dass n eine Zweierpotenz ist).

Die Grundidee ist, dass 2×2-Matrizen mit 7 statt 8 Skalarmultiplikationen multipliziert werden können:

Gegeben seien zwei Matrizen $A := \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ und $B := \begin{pmatrix} e & g \\ f & h \end{pmatrix}$.

Ziel ist, das Produkt

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot \begin{pmatrix} e & g \\ f & h \end{pmatrix} = \begin{pmatrix} r & s \\ t & u \end{pmatrix}$$

von A und B, also r, s, t und u, mit nur 7 Skalarmultiplikationen zu bestimmen. Dazu betrachtet man die sieben Produkte

$$\begin{aligned}
P_1 &:= a \cdot (g - h); \\
P_2 &:= (a + b) \cdot h; \\
P_3 &:= (c + d) \cdot e; \\
P_4 &:= d \cdot (f - e); \\
P_5 &:= (a + d) \cdot (e + h); \\
P_6 &:= (b - d) \cdot (f + h); \text{ und} \\
P_7 &:= (a - c) \cdot (e + g).
\end{aligned}$$

Dann gilt

$$\begin{aligned}
r &= P_5 + P_4 - P_2 + P_6; \\
s &= P_1 + P_2; \\
t &= P_3 + P_4; \text{ und} \\
u &= P_5 + P_1 - P_3 - P_7.
\end{aligned}$$

Analog kann man $n \times n$-Matrizen in vier $\frac{n}{2} \times \frac{n}{2}$-Matrizen unterteilen und die entsprechenden Untermatrizen berechnen. Strassens Algorithmus arbeitet also wie folgt:

Eingabe: Zwei $n \times n$-Matrizen A und B.
Ausgabe: Die Matrix $C := A \cdot B$.

(i) Teile A und B in $(n/2) \times (n/2)$-Matrizen, berechne mit Addition und Subtraktion die zu multiplizierenden Matrizen (also alle Faktoren, die in den Produkten $P_1, \dots, P_7$ vorkommen).

(ii) Führe 7 Multiplikationen von $(n/2) \times (n/2)$-Matrizen aus.

(iii) Berechne C durch geeignete Additionen bzw. Subtraktionen der Matrizen $P_1, \dots, P_7$.

Begründen Sie, warum der Algorithmus korrekt ist (Achtung: Matrizenmultiplikation ist nicht kommutativ).

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

Aufgabe 2 Rekursionsgleichungen *10 Punkte*

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

(a) $T(n) = T(9n/10) + n$;

(b) $T(n) = T(n - a) + T(a) + n$, für ein festes $a \geq 1$;

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

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

(e) $T(n) = 7T(n/3) + n^2$;

(f) $T(n) = 2T(n/2) + n \log n$; und

(g) $T(n) = T(n - 1) + 1/n$.

Aufgabe 3 Implementierung *10 Punkte*

(a) Implementieren Sie Karatsubas Algorithmus sowie die Schulmethode zur Multiplikation ganzer Zahlen. Vergleichen Sie die beiden Algorithmen experimentell. Ab welcher Eingabegröße ist Karatsuba besser?

(b) (*freiwillig, 5 Zusatzpunkte*) Entwickeln Sie eine hybride Implementierung, die bei kleineren Zahlen von Karatsuba auf die Schulmethode umschaltet, ähnlich wie bei Aufgabe 3 auf dem ersten Übungsblatt. Bestimmen Sie experimentell den günstigsten Umschaltpunkt.