

Algorithmen und Datenstrukturen (WS 2019)

Aufgabenblatt 2

zu bearbeiten bis: 18.11.2019 / 20.11.2019

Aufgabe 2.1 (O-Notation - Theorie)

Zeigen Sie mit Hilfe der Definition der O-Notation, dass gilt:

- $a \cdot n^2 \in O(n^3)$ für alle $a \in \mathbb{R}$.
- $a \cdot n^3 \notin O(n^2)$ für alle $a \in \mathbb{R}, a > 0$.

Aufgabe 2.2 (Matrix-Multiplikation - Theorie)

Matrizen sind zweidimensionale Arrays (in Java: z.B. `int[][]` oder `double[][]`). Die Matrix-Multiplikation ist folgendermaßen definiert: Multiplizieren wir zwei Matrizen A und B mit jeweils n Zeilen und Spalten, erhalten wir eine neue Matrix C (ebenfalls mit n Zeilen und Spalten). Wir berechnen einen Eintrag c_{ij} (in der i -ten Zeile und j -ten Spalte von C), indem wir das Skalarprodukt aus der i -ten Zeile von A und der j -ten Spalte von B berechnen. Dies ist hier illustriert:

$$\begin{array}{c} \begin{pmatrix} 1 & 2 & 3 \\ 4 & 1 & 0 \\ 5 & 2 & -3 \end{pmatrix} \times \begin{pmatrix} 2 & 1 & -3 \\ -1 & 0 & 4 \\ 1 & 2 & -4 \end{pmatrix} = \begin{pmatrix} 3 & 7 & -7 \\ 7 & 4 & -8 \\ 5 & -1 & 5 \end{pmatrix} \\ \underbrace{\hspace{10em}} \quad \underbrace{\hspace{10em}} \quad \underbrace{\hspace{10em}} \\ \mathbf{A} \qquad \qquad \mathbf{B} \qquad \qquad \mathbf{C} \\ \text{Zeile 2} \quad \times \quad \text{Spalte 1} = C_{21} \quad (=4 \times 2 + 1 \times (-1) + 0 \times 1) \end{array}$$

- Geben Sie die Gesamtzahl der benötigten arithmetischen Operationen (Multiplikationen und Additionen) für die Multiplikation zweier 3×3 -Matrizen an.
- Geben Sie nun die Gesamtzahl der benötigten Operationen in Abhängigkeit von n an. Zu welcher Komplexitätsklasse gehört Ihr Verfahren?

Aufgabe 2.3 (Matrix-Multiplikation - Praxis)

Wir betrachten nun die Praxis dieser theoretischen Erwägungen.

- a) Implementieren Sie die Multiplikation von Matrizen in Java. Messen Sie die Laufzeit für die Multiplikation zweier $n \times n$ -Matrizen für ($n = 100, 200, 400, 800, 1600$)
Sie können die Methode `runtime(int n)` verwenden, erhöhen Sie dafür den Timeout auf über 60s.
Wie sollte sich die Laufzeit entwickeln? Was beobachten Sie?
- b) Implementieren Sie die Funktion `transpose`, die eine $n \times n$ -Matrix A transponiert (Ergebnis: Matrix B). Das Element b_{ij} der transponierten Matrix ist das Element a_{ji} der ursprünglichen Matrix. Welche Komplexität hat die Funktion `transpose`?
- c) Implementieren Sie nun eine Variante der Matrix-Multiplikation welche die Matrix B vor der Multiplikation transponiert und bei der Multiplikation dann jeweils Zeilen von A mit Zeilen von B multipliziert (anstatt von Spalten). Messen Sie erneut die Laufzeit! Was beobachten Sie?