

Ausgabe am **21.10.2025**

Abgabe bis zum **28.10.2025 08:00 Uhr**



Prof. Dr. Thomas Wiemann

Algorithmen und Datenstrukturen

Übungsblatt 1

Wintersemester 2025/26

Hinweise zu den Übungen: Wir gehen davon aus, dass die Lösungen der Aufgaben in Java programmiert werden. Schriftliche Abgaben (z.B. die Tabelle unten) können Sie für die Präsentation z.B. als Bild, Skizze oder Konzept für ein Tafelbild vorbereiten. Zum Erreichen von Punkten bei der Besprechung der Aufgabe ist es nicht erforderlich, diese bis komplett zu lösen. Auch partielle Lösungen können gut bewertet werden, wenn klar ist, dass Sie sich vorbereitet haben. Wichtig ist, dass Sie sich so weit mit der Aufgabe beschäftigt haben, dass in der Übung sinnvoll über den Lösungsansatz und aufgetretene Probleme diskutiert werden kann, sodass eine Lösung / Lösungsskizze erarbeitet werden kann. Maximal gibt es 5 Punkte pro Präsentation. In das Portfolio können 2 Präsentationen im Umfang von max. 10 Punkten eingebracht werden.

Anmerkung zu den Sortieraufgaben: Hier sollen die Algorithmen qualitativ untersucht werden. Eine Betrachtung des Komplexitätsverhaltens erfolgt dann in der Vorlesung.

Aufgabe 1.1 (Laufzeitmessungen Suche)

Implementieren Sie lineare Suche, d.h., Suche nach einer ganzen Zahl in einem Array ganzer Zahlen, und binäre Suche (Suche nach der ersten '0') in einer Binärzahl in Java und führen Sie Laufzeitmessungen durch (in echter Zeit, nicht in Anzahl Schritten). Variieren Sie dabei die Eingabegrößen und die Resultate (Element vorhanden, ganz vorne, ganz hinten) und vergleichen Sie die echte Laufzeit der beiden Implementierungen. Entspricht das Ergebnis Ihren Erwartungen?

Aufgabe 1.2 (Laufzeitmessungen BUBBLE_SORT)

Machen Sie sich mit dem Sortieralgorithmus BUBBLE_SORT vertraut und implementieren Sie diesen in Java. Messen Sie analog zur ersten Aufgabe auch hier die echte Laufzeit. Entspricht das Ergebnis Ihren Erwartungen?

Aufgabe 1.3 (Laufzeitmessungen SELECTION_SORT)

Machen Sie sich mit dem Sortieralgorithmus SELECTION_SORT vertraut und implementieren Sie diesen in Java. Messen Sie analog zur ersten Aufgabe auch hier die echte Laufzeit. Entspricht das Ergebnis Ihren Erwartungen?

Aufgabe 1.4 (Wachstumsverhalten wichtiger Komplexitätsklassen)

Die unten stehende Tabelle spannt die Werte für die Anzahl der sich ergebenden Schritte für Eingabegrößen n und wichtiger Komplexitätsklassen auf. Füllen Sie die Tabelle mit Werten für die zu erwartende Laufzeit, unter der Annahme das ein Schritt $1 \mu\text{s}$ ($0,000001 \text{ s}$) dauert. Wandeln sie die Ergebnisse in sinnvolle Einheiten, z.B., Millisekunden, Sekunden, Jahre, Jahrhunderte um, so dass die Einträge intuitiv lesbar werden. Sollten die Werte für Ihre Rechenumgebung zu groß werden, tragen Sie ∞ ein.

$n =$	10	20	30	40	50	60
$\log(n)$						
n						
$n \cdot \log(n)$						
n^2						
n^3						
2^n						
$n!$						