

**Prüfung zu Lehrveranstaltung
708.031 Datenstrukturen und Algorithmen
GRUPPE B**

Es sind keinerlei Unterlagen oder Hilfsmittel erlaubt. Es dürfen nur einzelne, lose Blätter verwendet werden! Auf jedem Blatt muss der Name und die Matrikelnummer angegeben werden! Reine Arbeitszeit beträgt 90 Minuten. *Viel Erfolg!*

1. Asymptotische Schranken (8 Punkte)

Achtung: Antworten ohne Begründung werden **NICHT** berücksichtigt. Beweisen oder widerlegen Sie folgende Aussagen:

- a.) Eine Algorithmus, der eine Laufzeit $T(n) = \Theta(\log_b n)$ besitzt, hat immer einen Speicherbedarf von $S(n) = \Omega(\log_b n)$ für $b > 1$.
- b.) Es kann kein Suchalgorithmus existieren, welcher eine Laufzeit von $O(n)$ besitzt.
- c.) Es existiert ein Algorithmus, welcher eine Laufzeit von $O(n \cdot \log_3 n)$ besitzt und einen Speicher von $\Omega(n^2)$ benötigt.
- d.) Jeder Algorithmus mit einer Laufzeit $T(n) = \Theta(c \cdot n^c)$ mit $c > \pi$ ist auch $T(n) = \Omega(c \cdot \log_c n)$.

2. Hashtabellen (12 Punkte)

- a.) Erklären Sie die Begriffe und ihre Verwendung: ideale Hashfunktion, Belegungsfaktor, Double Hashing
- b.) Begründen Sie, ob folgende Hashfunktionen geeignet sind, wobei w der Schlüssel und m die Größe der Hashtabelle ist:
 - i.) $h(w) = (w + 1) \bmod m$
 - ii.) $h(w) = w * \log(100 \bmod m)$
 - iii.) $h(w) = \lfloor m \cdot (\lceil \frac{w}{\pi} \rceil - \frac{w}{\pi}) \rfloor$
- c.) Welche Möglichkeiten gibt es zur Kollisionbehandlung. Erklären Sie diese ausführlich (mit Skizze).

3. Sortieralgorithmen (8 Punkte)

- a.) Schreiben Sie einen Pseudocode zu *Mergesort* und erklären Sie die Vorgangsweise des Algorithmus in klaren Worten. Welche Vorteile hat Mergesort gegenüber anderen Sortieralgorithmen.
- b.) Leiten Sie Schritt für Schritt die untere Laufzeitschranke für vergleichende Sortierverfahren ab.
- c.) Erklären Sie die Begriffe *stabil* und *in-place* in Bezug auf Sortieralgorithmen und nennen Sie je einen Algorithmus, welche diese Eigenschaften besitzt.

4. (2,4)-Bäume (12 Punkte)

- a.) Fügen Sie folgende Werte in einen (2,4)-Baum ein: 2, 35, 40, 24, 31, 67, 55, 38, dann 67 entfernen und 62, 12 einfügen und 35 entfernen. Zeichnen Sie **jeden** einzelnen Schritt und kommentieren Sie diese.
- b.) Leiten Sie die Baumhöhe in Abhängigkeit der Anzahl der Elemente in einem (2,4)-Bäumen ab.
- c.) Zeigen Sie wie man effektiv Sortieren mit (2,4)-Bäumen implementieren kann (**Pseudocode, Laufzeit und Beschreibung**). Zeigen Sie, wieso dieser Sortieralgorithmus adaptiv ist.