

Name		Matr.-Nr.:	
Σ	Note:		

DI Stefan Klampff
08.02.2011 – Gruppe D

Prüfung zur Lehrveranstaltung 708.031 Datenstrukturen und Algorithmen

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.

1. Asymptotische Schranken (10 Punkte)

- a.) Definieren Sie in einer mathematischen Formulierung sowohl die O -Notation als auch die Θ -Notation. Erklären Sie Ihre Definitionen jeweils auch mit eigenen Worten anhand einer Skizze.
- b.) Lösen Sie die folgende rekursive Zeitgleichung durch iteratives Einsetzen: $T(n) = 3T(\frac{n}{3}) + n$, mit $T(1) = O(1)$.
- c.) Beweisen oder widerlegen Sie folgende Aussagen (Antworten ohne richtige Begründung erhalten **keine** Punkte!):
 - i.) Es existiert ein Algorithmus, welcher eine Laufzeit von $\Omega(2^n)$ und $O(n^3)$ besitzt.
 - ii.) Die Funktion $g(n) = n \cdot |\sin n|$ ist $\Omega(n)$.

2. Gestreute Speicherung (10 Punkte)

- a.) Erklären Sie das Grundprinzip der gestreuten Speicherung (mit Skizze). Was ist der Belegungsfaktor α ? Wann kommt es zu Kollisionen und wie können diese behandelt werden? Erklären Sie kurz die zwei in der Vorlesung besprochenen Methoden.
- b.) Erklären Sie die Divisions- und Multiplikationsmethode. Geben Sie die Bedingung für eine ideale Hashfunktion an.
- c.) Leiten Sie den **erwarteten** Suchaufwand in einer Hashtabelle T mit Überlauferlisten her. Betrachten Sie dabei getrennt die Fälle (i) $w \notin T$ und (ii) $w \in T$. Wie groß ist der Suchaufwand im worst-case und warum?

3. Finden der i-kleinsten Zahl (10 Punkte)

- a.) Erklären Sie ausführlich mit eigenen Worten das Prinzip von *Partition* (Zerlegen von Feldern) und die Anwendung für das Finden der i-kleinsten Zahl. Welche Verbesserung bringt eine *randomisierte* Version von *Partition*?
- b.) Leiten Sie die Laufzeit für das Finden der i-kleinsten Zahl für den besten Fall und den schlechtesten Fall her. Wie lautet die durchschnittliche Laufzeit?
- c.) In welchem bekannten Sortieralgorithmus wird ebenfalls *Partition* verwendet? Ist dieser Algorithmus *worst-case optimal*? Begründen Sie Ihre Antwort.

4. Richtig oder Falsch (10 Punkte)

Stimmen die folgenden Aussagen? Beachten Sie, dass es nur bei richtiger Antwort **mit** richtiger Begründung Punkte gibt.

- a.) *MergeSort* arbeitet *in-place*.
- b.) Das Minimum in einem in symmetrischer Reihenfolge sortierten Binarbaum kann immer in $O(h)$ Zeit gefunden werden (h ist die Höhe des Baums).
- c.) Der Codebaum für einen optimalen binären präfix-freien Code ist immer ein vollständiger, ausgeglichener Binärbaum.
- d.) *FastSearch* ist ordnungsmäßig manchmal schneller als die Interpolationssuche.
- e.) (2-4)-Bäume haben eine garantierte Höhe von $h = \Theta(\log n)$ ist (n ist die Anzahl der Blätter).

Viel Erfolg!