Name		Matr.-Nr.:	
$\sum$	Note:		

DI Stefan Klampfl
29.04.2010

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 **eigenen**, klaren Worten **und** einer mathematischen Formulierung sowohl die O-Notation als auch die Ω-Notation (mit Skizze!).

b.) Lösen Sie die folgende rekursive Zeitgleichung durch iteratives Einsetzen: $T(n) = 3T(\frac{n}{3}) + n^3$, mit $T(1) = O(1)$ (Hinweis: $\sum_{i=0}^{\infty} q^i = \frac{1}{1-q}$ für $|q| < 1$).

c.) Beweisen oder widerlegen Sie folgende Aussagen (Antworten ohne richtige Begründung erhalten **keine** Punkte!):

i.) Jeder Algorithmus mit einer Laufzeit $T(n) = \Theta(n^c)$ mit $c > 2$ ist auch $T(n) = \Omega(\log_c n)$.

ii.) Die Funktion $g(n) = n^{1+\sin n}$ ist $\Theta(n)$.

2. Suchen (10 Punkte)

a.) Erklären Sie das Prinzip der Interpolationssuche. Wie lauten die Laufzeiten im durchschnittlichen Fall und im schlimmsten Fall? Geben Sie ein Beispiel für den schlimmsten Fall an.

b.) Schreiben Sie einen Pseudocode fur die Binärsuche (mit Erklärung!) und bestimmen Sie seine Laufzeit.

c.) Erklären Sie das Prinzip von FastSearch.

3. Binärbäume (10 Punkte)

a.) Erklären Sie die Begriffe *symmetrische Reihenfolge*, *Hauptreihenfolge* und *Nebenreihenfolge*. In welcher Reihenfolge sind binäre Suchbäume sortiert?

b.) Schreiben Sie Pseudocodes zu den Funktionen *Minimum* und *Vorgänger* in binären Suchbäumen. Erklären Sie die einzelnen Schritte und geben Sie eine Laufzeitschranke an.

c.) Geben Sie zwei Möglichkeiten an, wie man eine Baumhöhe garantieren kann, die $O(\log n)$ ist (n ist die Anzahl der Knoten).

4. Richtig oder Falsch (10 Punkte)

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

a.) Es existiert ein optimaler, präfix-freier Binärcode mit den Kodewortlängen 2, 2, 3, 3, 3, 4, 5.

b.) Alle vergleichenden Sortierverfahren haben im worst case eine Laufzeit von $\Omega(n \log n)$.

c.) In einer Hashtabelle mit Überläuferlisten kommt es zu keiner Kollision, solange $\alpha < 1$.

d.) (2-4)-Bäume haben eine garantierte Höhe von $\Theta(n)$.

e.) Jedes absteigend vorsortierte Feld ist eine Halde, aber nicht umgekehrt.

Viel Erfolg!