

Name		Matr.-Nr.:	
Σ	Note:		

DI Stefan Klampfl
10.06.2010

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 die Θ -Notation (mit Skizze!).
- b.) Lösen Sie die folgenden rekursiven Zeitgleichungen durch iteratives Einsetzen:
 - i.) $T(n) = T(\frac{n}{2}) + n$, mit $T(1) = O(1)$
 - ii.) $T(n) = T(\sqrt{n}) + O(1)$, mit $T(2) = O(1)$
- c.) Beweisen oder widerlegen Sie folgende Aussagen (Antworten ohne richtige Begründung erhalten **keine** Punkte!):
 - i.) Es existiert ein Algorithmus, welcher eine Laufzeit von $\Omega(\log n)$ und $O(\sqrt{n})$ besitzt.
 - ii.) Die Funktion $g(n) = 2^{2^n}$ ist $O(2^n)$.

2. Sortieren (10 Punkte)

- a.) Erklären Sie ausführlich mit eigenen Worten das Prinzip von *Partition* (Zerlegen von Feldern) und die Anwendung in QuickSort. Welche Verbesserung bringt eine *randomisierte* Version von QuickSort?
- b.) Leiten Sie die Laufzeit von QuickSort für den besten Fall und den schlechtesten Fall her.
- c.) Was bedeuten die Begriffe *adaptiv* und *in-place* im Zusammenhang mit Sortierverfahren?

3. (2-4)-Bäume (10 Punkte)

- a.) Beschreiben Sie die Datenstruktur der (2-4)-Bäume. Worin liegt der Vorteil dieser Datenstruktur? Zeigen Sie, dass die Höhe eines (2-4)-Baums $h = \Theta(\log n)$ ist (n ist die Anzahl der Blätter).
- b.) Zeigen Sie, wie man (2-4)-Bäume zum **adaptiven** Sortieren verwenden kann. Erklären Sie, warum dieser Sortieralgorithmus adaptiv ist.
- c.) Welche Funktionen benötigt man, um eine **mischbare** Warteschlange mit Prioritäten zu implementieren? Erklären Sie, wie diese Datenstruktur als (2-4)-Baum implementiert werden kann, und geben Sie die Laufzeiten der Funktionen an.

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 ist möglich, dass es in einer Hashtabelle mit Überläuferlisten beim Einfügen zu keiner Kollision kommt, obwohl $\alpha \geq 1$.
- b.) Die Binärsuche ist immer langsamer als die Interpolationssuche.
- c.) Der Codebaum für einen optimalen binären präfix-freien Code ist immer ein vollständiger Binärbaum.
- d.) Alle vergleichenden Sortierverfahren haben im worst case eine Laufzeit von $\Omega(n \log n)$.
- e.) Bei einem Binärbaum, der in symmetrischer Reihenfolge sortiert ist, steht das Maximum immer in der Wurzel.

Viel Erfolg!