

**Prüfung zu Lehrveranstaltung
708.031/032 Datenstrukturen und Algorithmen,
WS 2005/06**

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. Lösen Sie folgende Rekursionen auf:

1. $T(n) = O(1) + T(\frac{n}{2})$
2. $T(n) = O(n) + T(n - 2)$
3. $T(n) = O(\sqrt{n}) + T(\sqrt{n})$

2. Nehmen Sie an, dass die Elemente eines linearen Feldes $A[1..n]$ schon folgendermaßen teilsortiert sind:

$$A[1] \geq A[2] \geq \dots \geq A[k]$$

$$A[k+1] \leq A[k+2] \leq \dots \leq A[n]$$

Des weiteren sei $A[k] \geq A[n]$ wobei k *unbekannt* ist.

- Wie lange benötigt MERGESORT *ordnungsmäßig* zum Sortieren dieses Feldes in Abhängigkeit von n und k ?
- Wie lange benötigt INSERTION-SORT *ordnungsmäßig* zum Sortieren dieses Feldes in Abhängigkeit von n und k ?
- Geben Sie einen Algorithmus an (Pseudocode), der das Feld in linearer Zeit sortiert.

3. Gestreute Speicherung (Hashing)

- Definieren Sie die Datenstruktur Hashtabelle und erklären Sie in *klaren* Worten Ihre Anwendung.
- Welche Methoden gibt es zur Behandlung von Kollisionen? Beschreiben Sie diese ausführlich.

4. Algorithmus-Entwurf

Die "Unbalanziertheit" $u(k)$ eines Knotens k in einem Binärbaum ist der Höhenunterschied der beiden darunterliegenden Teilbäume. Die "Unbalanziertheit" $U(B)$ eines Binärbaums B ist die maximale Unbalanziertheit aller Knoten k im Baum B .

Entwerfen Sie einen (möglichst einfachen) Algorithmus, der die Unbalanziertheit $U(B)$ eines Binärbaumes B mit n Knoten in linearer Zeit $O(n)$ berechnet.

Beschreiben Sie ihren Algorithmus Schritt für Schritt mit einem Pseudocode UND erklärenden Worten. Analysieren Sie sowohl den Zeit- als auch Speicherbedarf ($T(n)$ und $S(n)$).

Viel Erfolg!