

**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. Optimales Kodieren

- a.) Erklären Sie den Begriff präfix-frei.
- b.) Erklären Sie **Schritt für Schritt** die Konstruktion eines optimalen Kodierungsbaums nach **Huffman**.
- c.) Wie und mit welcher Datenstruktur wird der Algorithmus effektiv umgesetzt?

2. Pseudocode:

Gegeben ist folgender Pseudocode, wobei als Input ein Knoten k eines Binärbaums dient. Der Aufruf erfolgt mit $ALGO('Wurzel\ eines\ Baums')$:

```
ALGO (k)
1: S = 0
2: IF (LINKS(k)≠nil) AND (RECHTS(k)=nil) THEN
3:   S = 1 + ALGO(LINKS(k))
4: ELSEIF (LINKS(k)=nil) AND (RECHTS(k)≠nil) THEN
5:   S = 1 + ALGO(RECHTS(k))
6: ELSEIF (LINKS(k)≠nil) AND (RECHTS(k)≠nil) THEN
7:   S = ALGO(LINKS(k)) + ALGO (RECHTS(k))
8: RETURN S
```

- a.) Erklären Sie ausführlich **Schritt für Schritt** den Pseudocode und analysieren Sie die Laufzeit $T(n)$ in Abhängigkeit der Anzahl der Knoten n .
- b.) Welche Bedeutung hat der Rückgabewert S von $ALGO(w)$, wenn w die Wurzel eines nicht leeren Baums ist?
- c.) Zeichnen Sie einen Baum mit 15 Knoten, welcher für diesen Algorithmus einen Rückgabewert von $S = 0$ liefert.
- d.) Zeichnen Sie einen Baum mit 15 Knoten, für den der Algorithmus als Rückgabewert eine größtmögliche Zahl liefert.

3. Asymptotische Schranken:

- a.) Definieren Sie die Ω -Notation und erklären Sie sie mit eigenen Worten. Verwenden Sie eine Skizze zur Erklärung.
- b.) Geben Sie eine Funktion $f(n)$ an, die sowohl $f(n) \notin O(n)$ als auch $n \log n \notin O(f(n))$ erfüllt.
- c.) Wie verhalten sich die beiden Funktionsklassen $O(n^{\ln 5})$ und $O(5^{\ln n})$ zueinander? Beweisen Sie Ihre Aussagen.

Achtung: Weitere Aufgabe auf der Rückseite !!

4. Ein lineares Feld $A[1..n]$ soll so umgeordnet werden, dass im ersten Teil des Feldes alle negativen Zahlen, im mittleren Teil alle Nullen und im letzten Teil alle positiven Zahlen stehen. Eine korrekte Umordnung des Feldes $[3 -3 0 -4 1 0 4 4 0 0]$ ist zum Beispiel $[-4 -3 0 0 0 3 4 1 4]$.
Entwerfen Sie einen Algorithmus, der eine solche Umordnung durchführt, **ohne zusätzlichen Speicher** (bis auf Indizes und einzelne Werte) zu verwenden. Die volle Punkteanzahl gibt es nur für Algorithmen mit Laufzeit $O(n)$.
Erklären Sie **genau** und **verständlich** die Funktionsweise des Algorithmus. Sie müssen keinen Pseudocode angeben.

Viel Erfolg!