Name		MatrkNr.:	
$\sum$	Note:		

GRUPPE A
24.02.2009

Prüfung zu 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)

Achtung: Antworten ohne Begründung werden **NICHT** berücksichtigt. Beweisen oder widerlegen Sie folgende Aussagen:

a.) Es existiert kein Algorithmus dessen Laufzeit nicht $O(\frac{logn}{log\pi})$ und nicht $\Omega(\frac{n}{log\pi}\log n)$ für $n > 5$ ist.

b.) Der Speicherbedarf eines Algorithmus mit der Laufzeit $T(n) = O(n^3 \log n)$ kann nicht $S(n) = O(n^2 \log n)$ sein.

c.) Es existiert ein Algorithmus welcher eine Laufzeit von $O(n^2)$ und $\Theta(n^{\frac{7}{2}})$ besitzt.

d.) Jeder Algorithmus mit einer Laufzeit $T(n) = \Theta(\frac{\pi}{c} \cdot n^{(c-1)})$ mit $c > \pi$ ist auch $T(n) = \Omega(\frac{1}{c} \cdot log_c n)$.

e.) Die Rekursion $T(n) = 9T(\frac{n}{10}) + n$ ist $O(n \log_4 n)$ (Annahme: $T(2) = O(1)$).

2. Suchen (10 Punkte)

a.) Erklären Sie **ausführlich** mit eigenen Worten das Prinzip der Interpolationssuche. Wie lauten die Laufzeiten im durchschnittlichen und im schlimmsten Fall?

b.) Nennen Sie ein Beispiel für den schlimmsten Fall und beweisen Sie die Laufzeit dafür (**Ableitung!**).

c.) Erklären Sie das Prinzip von Fastsearch in Worten und mit Skizze.

3. Halden (10 Punkte)

a.) Erklären das Verhaldungsprinzip (Skizze) und geben Sie eine enge Schranken für die Laufzeit an.

b.) Zeigen Sie wie man mithilfe von Halden eine Warteschlange mit Prioritäten implementieren kann (Pseudocodes) und geben Sie die Laufzeiten an.

c.) Geben Sie den Pseudocode für eine Implementierung der optimalen Huffmankodierung mit Halden an.

4. Richtig oder Falsch (10 Punkte)

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

a.) Es gibt Fälle von Auftrittswahrscheinlichkeiten, für denen kein opimaler, präfix-freier Binärcode existiert.

b.) Der randomisierte Quicksort ist kein adaptiver Sortieralgorithmus.

c.) Radixsort kann für beliebige Inputs mit einer Laufzeit von $O(n)$ sortieren.

d.) Die Divisionmethode implementiert eine ideale Hashfunktion.

e.) Das Maximum in einen in SR sortierten Binärbaums kann immer in $O(\log n)$ gefunden werden (n ist die Anzahl der Knoten).

Viel Erfolg!