Technische Universität Braunschweig

Institut für Programmierung und Reaktive Systeme

Algorithmen und Datenstrukturen

Prof. Dr. Ursula Goltz / Dr. Werner Struckmann

27. September 2006

Name:

Vorname:

Fachrichtung:

Matrikelnummer:

Kennnummer: **Note:**

Die Bearbeitungszeit beträgt 120 Minuten. Die Klausur besteht aus 6 Aufgaben. Sie haben die Klausur bestanden, wenn Sie mindestens 37 von 74 möglichen Punkten erreicht haben. Sie dürfen keine Unterlagen verwenden. Dies schließt Taschenrechner ein. Schreiben Sie mit Tinte oder Kugelschreiber in blauer oder schwarzer Farbe. Benutzen Sie für Ihre Lösungen die beigefügten Blätter.

Aufgabe	1	2	3	4	5	6	Σ
max. Punkte	8	10	10	14	12	20	74
Punkte							

- Bitte prägen Sie sich Ihre Kennnummer gut ein. Aus Datenschutzgründen wird das Klausurergebnis nur unter dieser Kennnummer bekannt gegeben. Aus den gleichen Gründen können Ergebnisse weder telefonisch noch per E-Mail mitgeteilt werden.
- Die Ergebnisse der Klausur können Sie ab *Freitag, dem 6. Oktober 2006,* auf der WWW-Seite zu dieser Veranstaltung erfahren.
- Ihre Klausur können Sie am *Freitag, den 6. Oktober 2006,* von 11:00 bis 13:00 Uhr im Raum 251 des Informatikzentrums einsehen.

Aufgabe 1: *(Binäre Suchbäume)* Angenommen, es wurden Zahlen zwischen 1 und 1000 in einem binären Suchbaum gespeichert. Dann wurde nach der Zahl 333 gesucht. Welche der folgenden Sequenzen können nicht durchlaufen worden sein?

a) $2, 255, 403, 390, 320, 348, 340, 333$.

b) $888, 111, 777, 666, 222, 555, 444, 333$.

c) $783, 153, 720, 245, 733, 300, 340, 333$.

d) $950, 850, 243, 499, 275, 375, 299, 333$.

Geben Sie für jede Sequenz einzeln an, ob sie durchlaufen worden sein kann oder nicht.

8 Punkte

Lösung zu Aufgabe 1:

a) Die Sequenz ist möglich.

b) Die Sequenz ist möglich.

c) Die Sequenz ist nicht möglich, da $720 < 733$ ist.

d) Die Sequenz ist möglich.

Aufgabe 2: *(Asymptotische Notation)* Bitte kreuzen Sie an. Für jede richtige Antwort erhalten Sie einen Punkt, für jede falsche Antwort wird ein Punkt abgezogen. Kein Kreuz bzw. zwei Kreuze bedeuten 0 Punkte. Die minimale Gesamtpunktzahl für diese Aufgabe beträgt 0 Punkte.

			wahr	falsch
$5n^2 + 50n + 500$	$=$	$O(n^2)$	$\surd$	
$5n^2 + 50n + 500$	$=$	$\Theta(n^2)$	$\surd$	
$5n^2 + 50n + 500$	$=$	$\Omega(n^2)$	$\surd$	
$5n^2 + 50n + 500$	$=$	$o(n^2)$		$\surd$
$5n^2 + 50n + 500$	$=$	$\omega(n^2)$		$\surd$
3^n	$=$	$O(2^n)$		$\surd$
$\log_2 2n + \log_2 3n$	$=$	$O(\log n)$	$\surd$	
$\sin^2(x)$	$=$	$O(1)$	$\surd$	
$\sqrt{n+1}$	$=$	$O(\sqrt{n})$	$\surd$	
2^{n+1}	$=$	$O(2^n)$	$\surd$	

10 Punkte

Aufgabe 3: *(Listen, Bäume, Hashverfahren und Graphen)* Bitte kreuzen Sie an. Für jede richtige Antwort erhalten Sie einen Punkt, für jede falsche Antwort wird ein Punkt abgezogen. Kein Kreuz bzw. zwei Kreuze bedeuten 0 Punkte. Die minimale Gesamtpunktzahl für diese Aufgabe beträgt 0 Punkte.

	wahr	falsch
Ein Keller ist eine Liste, auf die nur an einem Ende zugegriffen werden kann.	$\surd$	
Eine Schlange ist eine Liste, auf die nur an einem Ende zugegriffen werden kann.		$\surd$
Eine Deque ist eine Liste, auf die nur an einem Ende zugegriffen werden kann.		$\surd$
In einem binären Suchbaum kann das minimale Element in der Zeit $O(\log n)$ bestimmt werden.		$\surd$
In einem Rot-Schwarz-Baum haben alle Pfade von der Wurzel zu einem Blatt die gleiche Länge.		$\surd$
In einem B-Baum der Ordnung t, $t \geq 2$, besitzt jeder Knoten maximal $2t - 1$ Schlüssel.	$\surd$	
In einem Max-Heap befindet sich das größte Element stets an der Wurzel.	$\surd$	
Beim Open-Hashing-Verfahren können keine Kollisionen auftreten.		$\surd$
Die Adjazenzmatrix eines Graphen mit n Knoten und m Kanten besteht aus n Zeilen und m Spalten.		$\surd$
Jeder gerichtete azyklische Graph besitzt eine topologische Sortierung.	$\surd$	

10 Punkte

Aufgabe 4: *(Sortierverfahren, imperative Algorithmen)* Gegeben seien ein Feld ganzer Zahlen $a[0..n-1]$, $n \geq 1$, und der folgende imperative Algorithmus, der dieses Feld sortiert.

for $j \leftarrow n-2$ **downto** 0 **do**
    $(*)$;
    $k \leftarrow a[j]$;
    $i \leftarrow j+1$;
    while $i \leq n-1$ und $a[i] < k$ **do**
        $a[i-1] \leftarrow a[i]$;
        $i \leftarrow i+1$;
    end
    $a[i-1] \leftarrow k$;
end

a) Beschreiben Sie die diesem Algorithmus zugrunde liegende Idee.

b) Gegeben sei das Feld $a = [5, 3, 4, 0, 2, 1]$. Geben Sie für jeden Durchlauf der äußeren Schleife den Inhalt von a an der Stelle $(*)$ an.

c) Wie oft wird der Test $i \leq n-1$ und $a[i] < k$ der While-Schleife im ungünstigsten Fall ausgeführt. Wann tritt dieser Fall ein?

d) Ist der Sortieralgorithmus stabil? Begründen Sie Ihre Aussage.

14 Punkte

Lösung zu Aufgabe 4:

a) Es handelt sich um eine Variante von Insertionsort. Das Feld wird aufsteigend sortiert. Im Gegensatz zum Algorithmus der Vorlesung wird am oberen Ende des Felds begonnen.

b)
```
5 3 4 0 2 1
5 3 4 0 1 2
5 3 4 0 1 2
5 3 0 1 2 4
5 0 1 2 3 4
```

c) $\frac{1}{2}n^2 - \frac{1}{2}n = \Theta(n^2)$

d) Der Algorithmus ist stabil, da $a[i] < k$ getestet wird.

Aufgabe 5: *(AVL-Bäume)*

a) Geben Sie die Definition eines AVL-Baums an.

b) Fügen Sie die Zahlen 6, 4, 8, 2, 5 und 1 in dieser Reihenfolge mit dem in der Vorlesung vorgestellten Algorithmus in einen anfangs leeren AVL-Baum ein. Zeichnen Sie den Baum nach dem Einfügen jeder Zahl. Erläutern Sie ggf. notwendige Rotationen.

12 Punkte

Lösung zu Aufgabe 5:

a) Ein binärer Suchbaum ist ein *AVL-Baum*, wenn für jeden seiner Knoten p gilt: Die Höhe des linken Teilbaums von p unterscheidet sich von der Höhe des rechten Teilbaums von p um höchstens 1.

b) Die Elemente 6, 4, 8, 2 und 5 können wie bei binären Suchbäumen eingefügt werden. Beim Einfügen der 1 ist eine Rechtsrotation erforderlich (siehe Folien 6-38 ff., 1. Fall).

Aufgabe 6: *(Komplexität und Korrektheit von Algorithmen)*

a) Gegeben sei das folgende Programmstück:

$x \leftarrow 0;$
for $i \leftarrow -1$ **to** $n-2$ **do**
  for $j \leftarrow -5$ **to** $5 \cdot i - 1$ **do**
    $x \leftarrow x + 1;$
  end
end

Welchen Wert besitzt die Variable x nach Ausführung dieser Anweisung? Geben Sie den genauen Wert in Abhängigkeit von n ohne Verwendung des Summenzeichens Σ an.

b) Für die Laufzeit a_n eines rekursiven Algorithmus wurde in Abhängigkeit von der Problemgröße n die Rekurrenzgleichung

$$a_n = 6a_{n-1} - 8a_{n-2} + 2,\ a_0 = 2,\ a_1 = 6$$

gefunden. Lösen Sie die Rekurrenzgleichung.

c) Für die Laufzeit $T(n)$ eines Teile-und-Beherrsche-Algorithmus wurde in Abhängigkeit von der Problemgröße n die Rekurrenzgleichung

$$T(n) = T\left(\frac{n}{2}\right) + n$$

gefunden. Lösen Sie diese Gleichung mithilfe des Mastertheorems.

d) Gegeben seien der Algorithmus S

$t \leftarrow 1;$
$y \leftarrow 0;$
while $t \leq x$ **do**
  $t \leftarrow 2t;$
  $y \leftarrow y + 1;$
end
$y \leftarrow y - 1;$

und die Vorbedingung $x \geq 1$. Welchen Wert besitzt die Variable y nach Ausführung des Algorithmus? Formulieren Sie eine entsprechende Nachbedingung q. Geben Sie eine Schleifeninvariante an, die zum Nachweis der partiellen Korrektheit von S bezüglich der Vorbedingung $x \geq 1$ und der Nachbedingung q geeignet ist.

20 Punkte

Lösung zu Aufgabe 6:

a) Es sei $T(n)$ der Wert der Variablen x in Abhängigkeit von n. Dann gilt

$$T(n) = \begin{cases} \frac{5}{2}n(n-1), & n \geq 1, \\ 0, & n \leq 0. \end{cases}$$

b) $a_n = \frac{4}{3} \cdot 4^n + \frac{2}{3}$ für alle $n \geq 0$.

c) $f(n) = n$, $g(n) = n^{\log_b a} = n^0 = 1$. Da $f(n) = n = \Omega(n^{0+\varepsilon})$ mit $\varepsilon = 1 > 0$ gilt, ist f polynomial größer als g. Weil außerdem die Regularitätsbedingung $af\left(\frac{n}{b}\right) = \frac{n}{2} \leq cn$ für $c = \frac{1}{2}$ für alle n erfüllt ist, sind die Vorausetzungen von Fall 3 erfüllt, das heißt $T(n) = \Theta(f(n)) = \Theta(n)$.

d) Die Variable y enthält den ganzzahligen Logarithmus zur Basis 2 von x, das heißt $y = \lfloor \log_2(x) \rfloor$. Die Nachbedingung lautet daher $2^y \leq x < 2^{y+1}$. Eine geeignete Schleifeninvariante ist $t = 2^y \wedge t \leq 2x \wedge x \geq 1$.