Technische Universität Braunschweig

Institut für Programmierung und Reaktive Systeme

Algorithmen und Datenstrukturen

Prof. Dr. Ursula Goltz / Dr. Werner Struckmann

27. März 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 31 von 62 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	10	8	16	62
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 31. März 2006,* auf der WWW-Seite zu dieser Veranstaltung erfahren.
- Ihre Klausur können Sie am *Mittwoch, den 5. April 2006,* von 9:00 bis 11:00 Uhr und von 13:00 bis 15:00 Uhr im Raum 251 des Informatikzentrums einsehen.

Aufgabe 1: *(Binäre Suchbäume)* Fügen Sie die Zahlen 4, 2, 6, 5, 8, 10, 11 und 9 in dieser Reihenfolge in einen anfangs leeren binären Suchbaum ein.

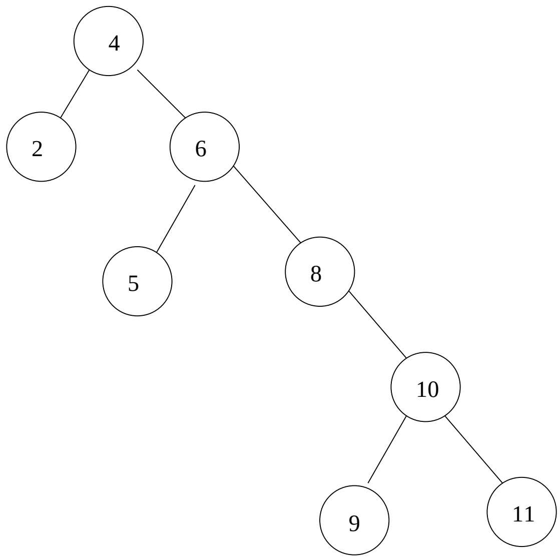
a) Zeichnen Sie den Baum nach dem Einfügen der Zahlen.

b) In welcher Reihenfolge werden die Zahlen bei einem Preorder-, Postorder- bzw. Inorder-Durchlauf ausgegeben?

8 Punkte

Lösung zu Aufgabe 1:

a) Der Baum hat die folgende Gestalt:



b) Preorder-Durchlauf: 4,2,6,5,8,10,9,11
Postorder-Durchlauf: 2,5,9,11,10,8,6,4
Inorder-Durchlauf: 2,4,5,6,8,9,10,11

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
$n^2 + n + 1 = O(n^2)$	$\surd$	
$n^2 + n + 1 = \Theta(n^2)$	$\surd$	
$n^2 + n + 1 = \Omega(n^2)$	$\surd$	
$n \log_2 n = O(n^2)$	$\surd$	
$n(n+1)/2 = O(n^2)$	$\surd$	
$(n+3)^k = \Theta(n^k)$	$\surd$	
$2^{n+2} = O(2^n)$	$\surd$	
$2^{2n} = O(2^n)$		$\surd$
$\sqrt{n} = O(\log n)$		$\surd$
$\log_2 n^2 = O(\log n)$	$\surd$	

10 Punkte

Aufgabe 3: *(Listen, Bäume 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
Keller arbeiten nach dem First-In-First-Out-Prinzip.		$\surd$
Ein Schlange ist eine Liste, bei der an einem Ende Elemente hinzugefügt und am anderen entfernt werden können.	$\surd$	
Für die Höhe h eines AVL-Baums gilt $h = O(\log n)$, wenn n die Anzahl der Knoten ist.	$\surd$	
Für einen Rot-Schwarz-Baum gilt: Wenn ein Knoten rot ist, so sind beide Kinder schwarz.	$\surd$	
In einem B-Baum haben alle Pfade von der Wurzel zu einem Blatt die gleiche Länge.	$\surd$	
Im ungünstigsten Fall liegt die Laufzeit von Heapsort in $O(n \log n)$.	$\surd$	
Die Adjazenzmatrix eines Graphen besitzt immer genauso viele Zeilen wie Spalten.	$\surd$	
Der Speicherbedarf eines Graphen $G = (V, E)$ mithilfe einer Adjazenzliste liegt in $O(\|V\| + \|E\|)$.	$\surd$	
Der Algorithmus von Prim bestimmt einen minimalen Spannbaum.	$\surd$	
Jeder gerichtete Graph besitzt eine topologische Sortierung.		$\surd$

10 Punkte

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

```
for i ← 1 to  n − 1 do
    p ← i;
    e ← a[i];
    for j ← i + 1 to  n do
        if a[j] ≥ e then
            p ← j;
            e ← a[j];
        end
    end
    t ← a[i];
    a[i] ← a[p];
    a[p] ← t;
    (∗)
end
```

a) Beschreiben Sie die diesem Algorithmus zugrunde liegende Idee.

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

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

10 Punkte

Lösung zu Aufgabe 4:

a) Für jedes i, $i = 1, \ldots, n-1$, bestimmt die innere Schleife das größte Element von $a[i]$,. . . ,$a[n]$ und tauscht es mit $a[i]$ (Selectionsort). Das Feld wird absteigend sortiert.

b) Der Algorithmus ist nicht stabil, da auf $\geq$ getestet wird.

c)

5	-2	0	4	1
5	4	0	-2	1
5	4	1	-2	0
5	4	1	0	-2

Aufgabe 5: *(Hashverfahren)* Gegeben seien die Schlüsselmenge $B = \{0, 1, \ldots, 50\}$, die Indexmenge $A = \{0, 1, \ldots, m-1\}$, $m = 7$, die Hashtabelle R mit den Feldern $R[0], \ldots, R[m-1]$ und die Hashfunktion $h : B \to A$ mit $h(k) = k \mod m$. Es sollen die Schlüssel

$$6, 10, 13, 2, 9, 16, 8, 7$$

in dieser Reihenfolge in die anfangs leere Hashtabelle eingefügt werden. Stellen Sie die Hashtabelle R nach Einfügen der Schlüssel dar. Die Kollisionsauflösung soll durch eine verkettete Liste erfolgen, wobei ein neues Element am Ende der Liste eingefügt wird.

8 Punkte

Lösung zu Aufgabe 5:

```
0:  7
1:  8
2:  2 -> 9 -> 16
3:  10
4:
5:
6:  6 -> 13
```

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

a) Gegeben sei das folgende Programmstück:

```
for i ← 2 to n − 1 do
    for j ← −1 to 3 · i + 1 do
        <a>
    end
end
```

Wie oft wird die Anweisung <a> ausgeführt? Geben Sie die genaue Anzahl $T(n)$ 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

$$a_n = 4a_{n-1} - 3a_{n-2},\ a_0 = 4,\ a_1 = 6,$$

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

$$T(n) = 4T\left(\frac{n}{2}\right) + n$$

gefunden. Lösen Sie diese Gleichung mithilfe des Mastertheorems.

d) Gegeben seien der Algorithmus S

```
i ← a;
x ← i;
while i < b do
    i ← i + 1;
    x ← x + i;
end
```

und die Vorbedingung $a \leq b$. Welchen Wert besitzt die Variable x 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 $a \leq b$ und der Nachbedingung q geeignet ist.

16 Punkte

Lösung zu Aufgabe 6:

a) $T(n) = \begin{cases} \frac{3}{2}n^2 + \frac{3}{2}n - 9, & n \geq 3, \\ 0, & n < 3. \end{cases}$

b) $a_n = 3^n + 3$, $n \geq 0$.

c) $T(n) = \Theta(n^2)$.

d) Für $a \leq b$ besitzt die Variable x den Wert $a + (a+1) + \ldots + b$. Als Nachbedingung q kann daher $x = \sum_{j=a}^{b} j \wedge i = b$ gewählt werden. Eine zum Nachweis der partiellen Korrektheit geeignete Schleifeninvariante ist $x = \sum_{j=a}^{i} j \wedge a \leq i \leq b$.