



186.172 Algorithmen und Datenstrukturen 1 VL 4.0

1. Übungstest WS 2010/11

26. November 2010

Machen Sie die folgenden Angaben bitte in deutlicher Blockschrift:

Nachname: Vorname:

Matrikelnummer: Studienkennzahl:

Anzahl abgegebener Zusatzblätter:

Legen Sie bitte Ihren Studentenausweis vor sich auf das Pult.

Sie können die Lösungen entweder direkt auf die Angabeblätter oder auf Zusatzblätter schreiben, die Sie auf Wunsch von der Aufsicht erhalten. Es ist nicht zulässig, eventuell mitgebrachtes eigenes Papier zu verwenden.

Die Verwendung von Taschenrechnern, Mobiltelefonen, Skripten, Büchern, Mitschriften, Ausarbeitungen oder vergleichbaren Hilfsmitteln ist unzulässig.

Die Arbeitszeit beträgt 55 Minuten.

| | A1: | A2: | A3: | Summe: |
|---------------------|----------------------|----------------------|----------------------|----------------------|
| Erreichbare Punkte: | 16 | 18 | 16 | 50 |
| Erreichte Punkte: | <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> |

Viel Erfolg!

Aufgabe 1.A: $\Omega/O/\Theta$ -Notation

(16 Punkte)

a) (6 Punkte)

Beweisen oder widerlegen Sie, dass die Beziehung $\sqrt{n} = \Theta(\sqrt[3]{n})$ gilt. Bedenken Sie, dass für einen Beweis gegebenenfalls auch geeignete Werte für die Konstanten c_1, c_2 und n_0 angegeben werden müssen.

b) (4 Punkte)

Seien $f(n)$, $g(n)$ und $h(n)$ Funktionen mit positivem Wertebereich. Kreuzen Sie in der folgenden Tabelle jeweils alle richtigen Folgerungen an. Jede Zeile wird nur dann gewertet, wenn sie vollständig richtig ist.

| Annahme | Folgerung: $f(n)$ ist in | | | |
|----------------------------------------------------------------------|--------------------------|-----------|----------------|-----------|
| | $\Omega(g(n))$ | $O(g(n))$ | $\Omega(h(n))$ | $O(h(n))$ |
| $(g(n) = \Omega(4 \cdot f(n))) \wedge (h(n) = \Omega(8 \cdot g(n)))$ | | | | |
| $(f(n) = O(g(n) + f(n))) \wedge (g(n) = \Omega(h(n)))$ | | | | |

c) (6 Punkte)

Bestimmen Sie die Laufzeit des unten angegebenen Codestückes in Abhängigkeit von n in Θ -Notation. *Hinweis:* Falls die Laufzeit für das gesamte Codestück nicht korrekt bestimmt wurde, werden auch Punkte für die Laufzeitbestimmungen der einzelnen Schleifen vergeben.

```

i = 1;
j = n;
k = 1;
while i < j do
    k = k + n;
    j = ⌊ $\frac{j}{2}$ ⌋;
end while
l = 1;
for i = 1, ..., k do
    for j = 1, ..., i do
        l = l + 1;
    end for
end for
j = n;
for i = 1, ..., l do
    j = 2 · j + i;
end for
    
```

Aufgabe 2.A: Sortierverfahren und Abstrakte Datentypen (18 Punkte)

a) (4 Punkte)

Sortieren Sie die folgenden binären Zahlen mit Quicksort:

$\langle 10, 11, 1, 0, 101, 111, 01, 00 \rangle$

Schreiben Sie dabei nach jedem Aufruf von Partition die entstandene Zahlenfolge auf und markieren Sie alle Elemente, die bereits ihre endgültige Position erreicht haben. Als Pivotelement soll jeweils das letzte Element der Teilfolge verwendet werden.

b) (14 Punkte)

Schreiben Sie in detailliertem Pseudocode eine Funktion $cutrange(L_1, x, y)$, die in einer aufsteigend sortierten doppelt verketteten zyklischen Liste L_1 alle Elemente löscht, deren Schlüssel $\geq x$ und $\leq y$ sind, und die entstandene Liste L_2 als Rückgabe der Funktion zurückliefert. Somit enthält diese Liste sortiert alle Schlüssel $< x$ und $> y$ aus der Liste L_1 .

Aufruf: $L_2 = cutrange(L_1, x, y)$;

Folgende Punkte sind weiters zu beachten:

- x, y und alle Schlüssel der Liste L_1 sind aus der Menge der reellen Zahlen.
- Es gilt $x \leq y$ und
- beim Aufruf des Algorithmus gilt: $L_1 \neq NULL$.
- Sollten sich alle Schlüssel der Liste L_1 innerhalb von x und y befinden, so soll der Aufwand des Algorithmus $\Theta(1)$ sein.
- Sollten sich alle Schlüssel der Liste L_1 außerhalb von x und y befinden, so soll der Aufwand des Algorithmus ebenfalls $\Theta(1)$ sein.

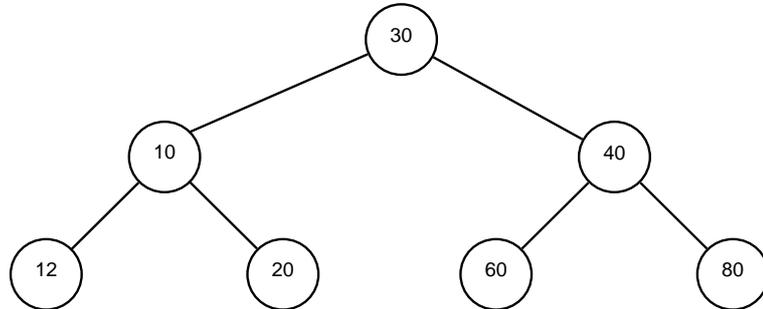
Geben Sie für Ihren Algorithmus den Aufwand für den Worst-Case und Best-Case in Θ -Notation in Abhängigkeit der Anzahl der Schlüssel n , die in der Liste gespeichert sind, an.

Aufgabe 3.A: Suchverfahren

(16 Punkte)

a) (8 Punkte)

Gegeben sei folgender binärer Baum t_1 :



- Geben Sie die Postorder Traversierung des binären Baumes t_1 an.
- Zeichnen Sie einen binären Suchbaum t_2 , der die Knoten und die gleiche Postorder Traversierung wie der binäre Baum t_1 hat.
- Handelt es sich bei ihrem binären Suchbaum t_2 um einen gültigen AVL-Baum? Begründen Sie ihre Antwort.

b) (8 Punkte)

Gegeben ist die Zahlenfolge

$\langle 33, 5, 15, 10, 12, 14, 13, 11 \rangle$.

- Fügen Sie alle Schlüssel der Folge nacheinander in einen anfangs leeren B-Baum der Ordnung 3 ein. Zeichnen Sie den resultierenden B-Baum und jeweils die B-Bäume, die direkt nach Aufspaltung eines Knotens entstehen. (die leeren Blätter können bei der Zeichnung entfallen).
- Fügen Sie alle Schlüssel der Folge nacheinander in einen anfangs leeren AVL-Baum ein. Zeichnen Sie den resultierenden AVL-Baum und alle Zwischenschritte, bei denen eine Reorganisation des Baumes notwendig ist. Geben Sie auch jeweils die Reorganisationsmaßnahme an.



186.172 Algorithmen und Datenstrukturen 1 VL 4.0

1. Übungstest WS 2010/11

26. November 2010

Machen Sie die folgenden Angaben bitte in deutlicher Blockschrift:

| | | | |
|-----------------|----------------------|------------------|--------------------------------------------------------|
| Nachname: | <input type="text"/> | Vorname: | <input type="text"/> |
| Matrikelnummer: | <input type="text"/> | Studienkennzahl: | <input type="text"/> |
| | | | Anzahl abgegebener Zusatzblätter: <input type="text"/> |

Legen Sie bitte Ihren Studentenausweis vor sich auf das Pult.

Sie können die Lösungen entweder direkt auf die Angabebblätter oder auf Zusatzblätter schreiben, die Sie auf Wunsch von der Aufsicht erhalten. Es ist nicht zulässig, eventuell mitgebrachtes eigenes Papier zu verwenden.

Die Verwendung von Taschenrechnern, Mobiltelefonen, Skripten, Büchern, Mitschriften, Ausarbeitungen oder vergleichbaren Hilfsmitteln ist unzulässig.

Die Arbeitszeit beträgt 55 Minuten.

| | | | | |
|---------------------|----------------------|----------------------|----------------------|----------------------|
| | B1: | B2: | B3: | Summe: |
| Erreichbare Punkte: | 16 | 16 | 18 | 50 |
| Erreichte Punkte: | <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> |

Viel Glück!

Aufgabe 1.B: Suchverfahren

(16 Punkte)

a) (8 Punkte)

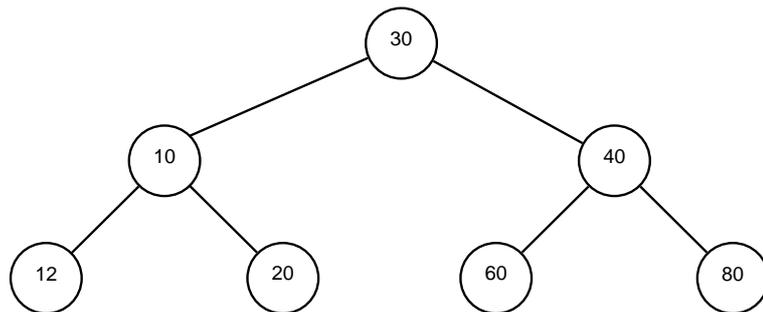
Gegeben ist die Zahlenfolge

$\langle 8, 18, 36, 13, 15, 17, 16, 14 \rangle$.

- Fügen Sie alle Schlüssel der Folge nacheinander in einen anfangs leeren B-Baum der Ordnung 3 ein. Zeichnen Sie den resultierenden B-Baum und jeweils die B-Bäume, die direkt nach Aufspaltung eines Knotens entstehen. (die leeren Blätter können bei der Zeichnung entfallen).
- Fügen Sie alle Schlüssel der Folge nacheinander in einen anfangs leeren AVL-Baum ein. Zeichnen Sie den resultierenden AVL-Baum und alle Zwischenschritte, bei denen eine Reorganisation des Baumes notwendig ist. Geben Sie auch jeweils die Reorganisationsmaßnahme an.

b) (8 Punkte)

Gegeben sei folgender binärer Baum t_1 :



- Geben Sie die Preorder Traversierung des binären Baumes t_1 an.
- Zeichnen Sie einen binären Suchbaum t_2 , der die Knoten und die gleiche Preorder Traversierung wie der binäre Baum t_1 hat.
- Handelt es sich bei ihrem binären Suchbaum t_2 um einen gültigen AVL-Baum? Begründen Sie ihre Antwort.

Aufgabe 2.B: $\Omega/O/\Theta$ -Notation

(16 Punkte)

a) (4 Punkte)

Seien $f(n)$, $g(n)$ und $h(n)$ Funktionen mit positivem Wertebereich. Kreuzen Sie in der folgenden Tabelle jeweils alle richtigen Folgerungen an. Jede Zeile wird nur dann gewertet, wenn sie vollständig richtig ist.

| Annahme | Folgerung: $f(n)$ ist in | | | |
|----------------------------------------------------------------------|--------------------------|----------------|-----------|----------------|
| | $O(g(n))$ | $\Omega(g(n))$ | $O(h(n))$ | $\Omega(h(n))$ |
| $(f(n) = O(g(n) + f(n))) \wedge (g(n) = \Omega(h(n)))$ | | | | |
| $(g(n) = \Omega(4 \cdot f(n))) \wedge (h(n) = \Omega(8 \cdot g(n)))$ | | | | |

b) (6 Punkte)

Bestimmen Sie die Laufzeit des unten angegebenen Codestückes in Abhängigkeit von n in Θ -Notation. *Hinweis:* Falls die Laufzeit für das gesamte Codestück nicht korrekt bestimmt wurde, werden auch Punkte für die Laufzeitbestimmungen der einzelnen Schleifen vergeben.

```

i = 1;
j = n;
k = 1;
while i < j do
    k = k + n2;
    j = ⌊ $\frac{j}{2}$ ⌋;
end while
l = 1;
for i = 1, ..., k do
    for j = 1, ..., i do
        l = l + 1;
    end for
end for
j = n;
for i = 1, ..., l do
    j = 2 · j + i;
end for
    
```

c) (6 Punkte)

Beweisen oder widerlegen Sie, dass die Beziehung $\sqrt{n} = \Theta(\sqrt[4]{n})$ gilt. Bedenken Sie, dass für einen Beweis gegebenenfalls auch geeignete Werte für die Konstanten c_1 , c_2 und n_0 angegeben werden müssen.

Aufgabe 3.B: Sortierverfahren und Abstrakte Datentypen (18 Punkte)

a) (14 Punkte)

Schreiben Sie in detailliertem Pseudocode eine Funktion $cutrange(L_1, x, y)$, die in einer aufsteigend sortierten doppelt verketteten zyklischen Liste L_1 alle Elemente löscht, deren Schlüssel $\geq x$ und $\leq y$ sind, und die entstandene Liste L_2 als Rückgabe der Funktion zurückliefert. Somit enthält diese Liste sortiert alle Schlüssel $< x$ und $> y$ aus der Liste L_1 .

Aufruf: $L_2 = cutrange(L_1, x, y)$;

Folgende Punkte sind weiters zu beachten:

- x, y und alle Schlüssel der Liste L_1 sind aus der Menge der reellen Zahlen.
- Es gilt $x \leq y$ und
- beim Aufruf des Algorithmus gilt: $L_1 \neq NULL$.
- Sollten sich alle Schlüssel der Liste L_1 innerhalb von x und y befinden, so soll der Aufwand des Algorithmus $\Theta(1)$ sein.
- Sollten sich alle Schlüssel der Liste L_1 außerhalb von x und y befinden, so soll der Aufwand des Algorithmus ebenfalls $\Theta(1)$ sein.

Geben Sie für Ihren Algorithmus den Aufwand für den Worst-Case und Best-Case in Θ -Notation in Abhängigkeit der Anzahl der Schlüssel n , die in der Liste gespeichert sind, an.

b) (4 Punkte)

Sortieren Sie die folgenden binären Zahlen mit Quicksort:

$\langle 11, 100, 10, 1, 110, 111, 010, 01 \rangle$

Schreiben Sie dabei nach jedem Aufruf von Partition die entstandene Zahlenfolge auf und markieren Sie alle Elemente, die bereits ihre endgültige Position erreicht haben. Als Pivotelement soll jeweils das letzte Element der Teilfolge verwendet werden.