Universität Dortmund
FB Informatik
Lehrstuhl XI / Algorithm Engineering

Prof. Dr. Petra Mutzel
Markus Chimani, Henrik Björklund, Carsten Gutwenger,
Karsten Klein, Michael Meier

Sommersemester 2006

DAP2 Übung – Test (Nachtest)

Datum: 19. Mai 2006 Gruppe: Violett

Matrikelnummer:	
Nachname:	
Vorname:	

Der Test dauert 45 Minuten. Es sind keine Unterlagen erlaubt. Es gibt 16 Punkte zu erreichen. Mit dem Beginn des Ausfüllens dieses Test gilt die Prüfungsfähigkeit als bestätigt. Schummeln jeglicher Art führt zu einer Bewertung mit 0 Punkten und dem Ausschluß von der aktiven Übung. **Leserlich schreiben!** Unlesbares wird als falsch gewertet.

Aufgabe 1. (1,5 Punkte)

Gib die Definition für $g(n) = o(f(n))$ an.

Aufgabe 2. (1 Punkt)

Vereinfache die folgenden Ausdrücke mit Hilfe der Regeln der O-Notationen.

$$\Theta(f(n)) \setminus \omega(f(n)) =$$

$$4n + \Theta(n^4) + \Omega(n^2) =$$

Aufgabe 3. (1,5 Punkte)

Welche Aussagen sind richtig? Kreuze an; es können jeweils auch mehrere Aussagen richtig sein:

$3n^{2k} =$	☐ $\Theta(2n^{3k})$	☐ $\omega(2n^{2k})$	☐ $O(3n^{3k})$
$n^{\frac{3}{7}} + \frac{7}{3}\log n =$	☐ $o(\sqrt{n})$	☐ $\Omega(\log n)$	☐ $\omega(n)$

Aufgabe 4. (2 Punkte)

Bestimme die Θ-Notation für $f(n) := 2n + \sqrt{n^3}$ und beweise sie formal.

Aufgabe 5. (2 Punkte)

Gegeben sei eine Folge von Werten $w_1, w_2, \ldots, w_n, w_{n+1}, \ldots, w_{2n}$. Benutze im Folgenden ausschließlich die Standardoperationen für Stacks bzw. Queues. Füge zunächst die ersten n Werte der Reihe nach in einen Stack S ein; danach füge die anderen n Werte in eine Queue Q ein. Anschliessend entferne abwechselnd jeweils ein Element aus diesen beiden Datenstrukturen (beginnend mit S) und füge sie in eine zweite Queue Z ein.

In welcher Reihenfolge erhält man die Werte wenn man sie aus Z entfernt? (Nur dies ist gesucht! Kein Code!)

Aufgabe 6. (1,5 Punkte)

Gib das Zwischenergebnis von Merge-Sort nach den ersten fünf Aufrufen von Merge an. Als Eingabefolge dient dabei:

$$T, P, O, L, S, T, E, R$$

Aufgabe 7. (1 Punkt)

Wir betrachten den ersten in der Vorlesung vorgestellten Quick-Sort Algorithmus (also ohne Modifikation für Randomisierung oder Speicherreduktion). Wie hoch ist der asymptotische Speicherbedarf im Best- und Worst-Case?

Aufgabe 8. (2 Punkte) Ist folgender Beweis korrekt? Wenn nein: worin liegt der Fehler?

> **Satz:** Sei n die Länge der zu sortierenden Folge. Jeder deterministische Sortieralgorithmus benötigt maximal $O(n!)$ Vergleiche.
>
> **Beweis:** Im Entscheidungsbaum entspricht jedes Blatt einer Eingabepermutation. Ein deterministischer Algorithmus terminiert bei gleicher Eingabefolge immer im selben Blatt. Wir haben daher einen Entscheidungsbaum mit $n!$ Blättern. Da jeder innere Knoten eine Entscheidung repräsentiert, enthält er jeweils ein linkes und ein rechtes Kind. Daher existieren maximal $n! - 1$ innere Knoten. Bei jeder Probleminstanz entspricht die Anzahl der Vergleiche der Anzahl der inneren Knoten auf dem Weg von der Wurzel zum korrespondierenden Blatt. Daher werden maximal $n! - 1$ Vergleiche durchgeführt.

Aufgabe 9. (1,5 Punkte)

Zeichne den natürlichen binären Suchbaum der durch diese Folge von Operationen entsteht (Der Baum sei anfangs leer):

Insert(G), Insert(C), Insert(E), Insert(D), Insert(L), Insert(H), Insert(I), Delete(G)

Die Elemente sind natürlich alphabetisch zu ordnen.

Aufgabe 10. (2 Punkte)

Gegeben ist ein vollständiger binärer Suchbaum mit den folgenden Elementen:

$$1, 5, 6$$

Zeichne den AVL-Baum der durch die folgende Folge von Operationen entsteht:

Insert(2), Insert(3), Insert(4)

Die Elemente sind natürlich gemäß den natürlichen Zahlen zu ordnen.

Viel Erfolg!

Universität Dortmund
FB Informatik
Lehrstuhl XI / Algorithm Engineering

Prof. Dr. Petra Mutzel
Markus Chimani, Henrik Björklund, Carsten Gutwenger,
Karsten Klein, Michael Meier

Sommersemester 2006

DAP2 Übung – Test (Nachtest)

Datum: 19. Mai 2006 Gruppe: Orange

Matrikelnummer:	
Nachname:	
Vorname:	

Der Test dauert 45 Minuten. Es sind keine Unterlagen erlaubt. Es gibt 16 Punkte zu erreichen. Mit dem Beginn des Ausfüllens dieses Test gilt die Prüfungsfähigkeit als bestätigt. Schummeln jeglicher Art führt zu einer Bewertung mit 0 Punkten und dem Ausschluß von der aktiven Übung. **Leserlich schreiben!** Unlesbares wird als falsch gewertet.

Aufgabe 1. (1,5 Punkte)

Gib die Definition für $g(n) = \omega(f(n))$ an.

Aufgabe 2. (1 Punkt)

Vereinfache die folgenden Ausdrücke mit Hilfe der Regeln der O-Notationen.

$$\Theta(f(n)) \setminus o(f(n)) =$$

$$2n + \Omega(\log n) + O(\sqrt{n}) =$$

Aufgabe 3. (1,5 Punkte)

Welche Aussagen sind richtig? Kreuze an; es können jeweils auch mehrere Aussagen richtig sein:

$2n^{3k} =$ ☐ $\Omega(3n^{2k})$ ☐ $\Theta(2n^{2k})$ ☐ $o(3n^{3k})$

$n^{\frac{5}{9}} + \frac{9}{5}\log n =$ ☐ $O(n)$ ☐ $\Omega(n^{5,9})$ ☐ $\omega(\sqrt{n})$

Aufgabe 4. (2 Punkte)

Bestimme die Θ-Notation für $f(n) := 3n + \sqrt[3]{n^4}$ und beweise sie formal.

Aufgabe 5. (2 Punkte)

Gegeben sei eine Folge von Werten $w_1, w_2, \ldots, w_n, w_{n+1}, \ldots, w_{2n}$. Benutze im Folgenden ausschließlich die Standardoperationen für Stacks bzw. Queues. Füge zunächst die ersten n Werte der Reihe nach in einen Queue Q ein; danach füge die anderen n Werte in einen Stack S ein. Anschliessend entferne abwechselnd jeweils ein Element aus diesen beiden Datenstrukturen (beginnend mit Q) und füge sie in einen zweiten Stack Z ein.

In welcher Reihenfolge erhält man die Werte wenn man sie aus Z entfernt? (Nur dies ist gesucht! Kein Code!)

Aufgabe 6. (1,5 Punkte)

Gib das Zwischenergebnis von Merge-Sort nach den ersten fünf Aufrufen von MERGE an. Als Eingabefolge dient dabei:

$$T, H, M, U, S, T, E, R$$

Aufgabe 7. (1 Punkt)

Wir betrachten den ersten in der Vorlesung vorgestellten Quick-Sort Algorithmus (also ohne Modifikation für Randomisierung oder Speicherreduktion). Wodurch (durch welche Datenstruktur) wird der nicht-konstante Speicherbedarf des Algorithmus verursacht?

Aufgabe 8. (2 Punkte) Ist folgender Beweis korrekt? Wenn nein: worin liegt der Fehler?

Satz: Sei n die Länge der zu sortierenden Folge. Jeder deterministische Sortieralgorithmus benötigt maximal $O(n!)$ Vergleiche.

Beweis: Im Entscheidungsbaum entspricht jedes Blatt einer Eingabepermutation. Ein deterministischer Algorithmus terminiert bei gleicher Eingabefolge immer im selben Blatt. Wir haben daher einen Entscheidungsbaum mit $n!$ Blättern. Da jeder innere Knoten eine Entscheidung repräsentiert, enthält er jeweils ein linkes und ein rechtes Kind. Daher existieren maximal $n! - 1$ innere Knoten. Bei jeder Probleminstanz entspricht die Anzahl der Vergleiche der Anzahl der inneren Knoten auf dem Weg von der Wurzel zum korrespondierenden Blatt. Daher werden maximal $n! - 1$ Vergleiche durchgeführt.

Aufgabe 9. (1,5 Punkte)

Zeichne den natürlichen binären Suchbaum der durch diese Folge von Operationen entsteht (Der Baum sei anfangs leer):

Insert(4), Insert(7), Insert(10), Insert(2), Insert(9), Insert(3), Insert(1), Delete(4)

Die Elemente sind natürlich gemäß den natürlichen Zahlen zu ordnen.

Aufgabe 10. (2 Punkte)

Gegeben ist ein vollständiger binärer Suchbaum mit den folgenden Elementen:

$$A, B, F$$

Zeichne den AVL-Baum der durch die folgende Folge von Operationen entsteht:

Insert(E), Insert(D), Insert(C)

Die Elemente sind natürlich alphabetisch zu ordnen.

Viel Erfolg!