

Klausur: Algorithmen und Datenstrukturen

Datum: 14.02.2007
Bearbeitungszeit: 120 Minuten
Erlaubte Hilfsmittel: Taschenrechner

Name:

Matr.-Nr.:

A1	A2	A3	A4	A5	A6	A7	A8	Gesamt	Note
7	18	16	9	13	20	10	7	100	

Hinweise:

- Vergewissern Sie sich, dass alle 5 Seiten der Aufgabenstellung vorhanden sind.
- Bitte schreiben Sie Ihren Namen und/oder Ihre Matrikelnummer auf dieses Blatt.
- Die Aufgabenblätter sind am Ende der Klausur abzugeben.
- Falls Sie der Meinung sind, dass eine Aufgabe unklar oder zweideutig formuliert wurde, treffen Sie sinnvolle Annahmen, dokumentieren Sie diese und bearbeite Sie die Aufgabe entsprechend.
- Täuschungsversuche führen zum Ausschluss von der Klausur.

Viel Glück!

Aufgabe 1: Komplexität

(maximale Punktzahl: 7)

1. Bestimmen Sie die jeweilige Komplexitätsklasse und deren verbale Beschreibung (z.B. $O(1)$ – konstanter Aufwand) zu den folgenden Komplexitätsausdrücken:

$O(3n \log n)$, $O(n^{77})$, $O(2^{3n})$

2. Analysieren Sie folgende Algorithmen und bestimmen Sie deren Komplexitätsklasse in Abhängigkeit von n (in O-Notation)

- a.

```
for i:= 1 to (n · n) do
  a := 0
od
```
- b.

```
for i := 1 to n do
  for j := 1 to 2 do
    k := k+1
  od
od
```
- c.

```
x:= 87+20
if x > 100 then
  y := x
else
  for i := 1 to n do
    if a > y then y := a fi
  od
fi
```

Aufgabe 2: Sortieren

(maximale Punktzahl: 18)

1. Wie hoch ist der Aufwand für die Suche in einen ungeordneten Feld?
2. Wie hoch ist der Aufwand, wenn man ein ungeordnetes Feld erst mit *BubbleSort* sortiert und dann sucht?
3. Wie oft muss man suchen, damit der Aufwand für einmal Sortieren mit *BubbleSort* und m -mal Suchen besser ist, als m -mal im unsortierten Feld suchen? (Oder anders gefragt: Ab wann lohnt sich das Sortieren?)
4. Unter welchen Bedingungen hat das *QuickSort*-Verfahren seine schlechteste Komplexität? Welche Komplexität ist das?

5. Welche Komplexität hat das *SelectionSort*-Verfahren? Beweisen Sie Ihre Behauptung.

Aufgabe 3: Entwurf von Algorithmen

(maximale Punktzahl: 16)

Es ist folgende Menge von Gegenständen gegeben, die alle ein bestimmtes Gewicht (g) und einen bestimmten Wert (w) haben. Der Index (i) steht für eine laufende Nummerierung der Gegenstände.

Index i	1	2	3	4
Gewicht g	2	2	6	5
Wert w	6	3	5	4

Es soll ein LKW beladen werden, dessen maximale Kapazität 10 beträgt. Der LKW soll aber so beladen werden, dass die Summe der transportierten Gegenstände einen möglichst hohen Wert hat.

1. Welche *Algorithmenmuster* können Sie zur Lösung der Aufgabe verwenden? Beschreiben Sie die Muster in kurzen Stichpunkten, nennen Sie jeweils Vor- und Nachteile.
2. Geben Sie eine optimale Beladung für den LKW an. Welches Algorithmenmuster verwenden Sie dafür? Beschreiben Sie Ihr Vorgehen mit Stichpunkten.

Aufgabe 4: 2-3-4-Baum

(maximale Punktzahl: 9)

1. Was ist ein 2-3-4 Baum? (genauer: Welche Vorteile hat er im Vergleich zum Suchbaum? Welche Arten von Knoten hat er?)
2. Fügen Sie in folgender Reihenfolge die Zahlen 5, 1, 19, 25, 17, 21, 9, 15, 14 in einen vorher leeren 2-3-4 Baum ein

Aufgabe 5: Rot-Schwarz-Baum

(maximale Punktzahl: 13 Punkte)

1. Welche Eigenschaften hat ein Rot-Schwarz-Baum?
2. Fügen Sie die Zahlen 8, 5, 6, 7, 9, 10, 4 in einen anfangs leeren Rot-Schwarz-Baum ein. Zeichnen Sie den Baum nach jeder Einfügeoperation. Machen Sie dabei die notwendigen Umfärbungen und Rotationen deutlich.

Aufgabe 6: B-Baum

(maximale Punktzahl: 20 Punkte)

1. Welche Eigenschaften hat ein B-Baum?
2. Fügen Sie die Schlüssel 10, 21, 26, 7, 15, 20, 11, 14, 5, 12 und 16 in einen anfangs leeren B-Baum der Ordnung $m = 5$ ein. Zeichnen Sie den Baum nach jeder Einfügeoperation.
3. Könnte sich die Höhe verändern, wenn die obigen Schlüssel in einer anderen Reihenfolge eingegeben werden?
4. Wieviel Schlüssel können in einem B-Baum der Ordnung $m = 2$ und der Höhe 3 gespeichert werden? Geben Sie die minimale und maximale Anzahl an.
5. In Deutschland werden 5-stellige Postleitzahlen verwendet. Diese (maximal 100000 Werte) sollen in einem B-Baum gespeichert werden, die Höhe des Baumes soll 3 betragen. Welche Ordnung muss der B-Baum haben?

Aufgabe 7: Heap

(maximale Punktzahl: 10 Punkte)

1. Sortieren Sie die Zahlenfolge 5, 15, 1, 3, 19, 14, 0 mit HeapSort. Zeichnen Sie den Heap für jeden Sortierschritt jeweils nach dem die Heap-Eigenschaft wieder hergestellt ist. Erklären Sie das Verfahren kurz.
2. Vergleichen Sie die Komplexität von HeapSort mit der Komplexität von Merge-Sort.

Aufgabe 8: Suchbaum

(maximale Punktzahl: 7 Punkte)

1. Fügen Sie in einen anfangs leeren Suchbaum die Zahlen 2, 4, 1, 7, 0, 10, 8 in dieser Reihenfolge ein. Zeichnen Sie nur den fertigen Suchbaum (keine Zwischenschritte.)
2. Geben Sie die *levelorder*, *preorder* und *inorder*- Reihenfolge der Knoten an.