

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
10	10	17	20	8	15	10	10	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: 10)

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

$O(300)$, $O(3n \log n)$, $O(n^{77})$, $O(2^{3n})$

(? Punkte)

2. Analysieren Sie folgende Algorithmen und bestimmen Sie deren Komplexitätsklasse in Abhängigkeit von n (in O-Notation) (4 Punkte)

a. for i:= 1 to n do

 a := 0

od

b. for i := 1 to n do

 for j := 1 to n do

 k := k+1

 od

od

c. 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: 10)

1. Gegeben sei eine Folge von 6 Schlüsseln: 7, 3, 1, 6, 5, 9. Sortieren Sie diese Folge mit dem *MergeSort*-Verfahren und dem *QuickSort*-Verfahren. Erstellen Sie für jedes Verfahren einen Trace von Hand, d.h. dokumentieren Sie jede Veränderung an der Schlüsselfolge, die sich während der Sortierung (nach einer Iteration des Sortierverfahrens) ergibt.

Beschreiben Sie die Arbeitsweise beider Verfahren (Stichpunkte). (8 Punkte)

2. Nennen Sie die Komplexitätsklassen (O-Notation) für die Sortierverfahren: MergeSort, QuickSort, BubbleSort, SelektionSort. (2 Punkte)

Aufgabe 3: B-Baum

(maximale Punktzahl: 17)

1. Fügen Sie die Schlüssel 7, 19, 23, 4, 12, 17, 8, 11, 2, 9 und 13 in einen anfangs leeren B-Baum der Ordnung $m = 5$ ein. Zeichnen Sie nach jedem Einfügevorgang den Baum neu. (7 Punkte)
2. Welche Höhe hat der Baum? Wieviele Schlüssel könnte dieser Baum maximal aufnehmen? (2 Punkte)
3. Löschen Sie der Reihe nach die Schlüssel 12, 2 und 7 aus dem Baum. Zeichnen Sie den Baum nach jeder Löschung und erklären Sie (Stichpunkte, kurz!) Ihr Vorgehen. (5 Punkte)
4. Ein Warenhaus handelt mit 75.000 Artikeln. Die Artikelnummern aller Artikel sollen in einem B-Baum gespeichert werden. Die Höhe des Baumes soll 3 betragen. Welche Ordnung hat der B-Baum? (3 Punkte)

Aufgabe 4: Suchbaum

(maximale Punktzahl: 20)

1. Was ist ein binärer Suchbaum? Zeichnen Sie 5 verschiedene Suchbäume mit unterschiedlichen Höhen, die die Zahlen von 7 bis 13 beinhalten. Geben Sie die jeweilige Höhe an. (6 Punkte)
2. Fügen Sie die Buchstaben A, L, G, O, R, I, T, H, M, U, S in dieser Reihenfolge in einen anfangs leeren binären Suchbaum ein (es gilt die alphabetische Ordnung). Zeichnen Sie den Suchbaum nachdem Sie alle Buchstaben eingefügt haben. Geben Sie die *preorder*, *postorder* und *inorder*-Reihenfolge der Knoten an. (6 Punkte)
3. Löschen Sie aus dem Suchbaum, den Sie in Aufgabe 4.2 erstellt haben, den Knoten mit dem Wert L. Zeichnen Sie den resultierenden Suchbaum und begründen Sie Ihr Vorgehen. (4 Punkte)
4. In einen binären Suchbaum wurden Zahlen zwischen 1 und 1000 eingefügt. Gesucht werden soll die Zahl 363. Welche der unten notierten Folgen können keine Suchpfade sein? Begründen Sie ihre Entscheidung. (4 Punkte)
 - (a) 2, 252, 401, 398, 330, 344, 253, 363
 - (b) 924, 220, 911, 244, 898, 258, 362, 363
 - (c) 925, 202, 911, 240, 400, 245, 363
 - (d) 2, 399, 387, 219, 266, 382, 381, 260, 363
 - (e) 935, 278, 347, 621, 512, 392, 358, 363

Aufgabe 5: Rot-Schwarz-Baum

(maximale Punktzahl: 8 Punkte)

Fügen Sie die Zahlen $7, 5, 3, 8, 9, 10, 2$ 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: Hash

(maximale Punktzahl: 15 Punkte)

1. Gegeben ist folgende Hashtabelle

0	1	2	3	4	5	6
20	25	16	31	80	32	66

Diese ist unter Verwendung der Hashfunktion $h(x) = x \bmod 7$ und linearem Sondieren durch Eintragen der Elemente in einer gewissen Reihenfolge entstanden.

Geben Sie eine Reihenfolgen an, in denen die Elemente in die anfangs leere Hashtabelle eingefügt worden sein können. (7 Punkte)

2. Tragen Sie die Zahlen $10, 11, 16, 24, 29$ in dieser Reihenfolge in eine Hash-Tabelle der Größe 5 ein. Verwenden Sie dazu die Hashfunktion $h(x) = x \bmod 5$. Behandeln Sie Kollisionen durch quadratisches Sondieren. Geben Sie zu den vorgegebenen Zahlen die Hash-Werte an. Zählen Sie für jede eingefügte Zahl die Anzahl der Kollisionen. Zeichnen Sie die resultierende Hashtabelle. (8 Punkte)

Aufgabe 7: Heap

(maximale Punktzahl: 10 Punkte)

1. Welche Eigenschaften hat ein Heap? (1 Punkt)
2. Sortieren Sie die Zahlenfolge $7, 17, 3, 5, 21, 16, 1$ mit HeapSort. Zeichnen Sie den Heap für jeden Sortierschritt jeweils nach dem die Heap-Eigenschaft wieder hergestellt ist. Erklären Sie das Verfahren kurz. (8 Punkte)
3. Zu welcher Komplexitätsklasse gehört HeapSort? (1 Punkt)

Aufgabe 8: Graphen

(maximale Punktzahl: 10 Punkte)

Gegeben sei ein gerichteter Graph $G = (V, E)$ mit

$$V = \{1, 2, 3, 4, 5, 6, 7, 8\}$$

$$E = \{(1, 3), (1, 8), (3, 6), (3, 5), (4, 2), (4, 7), (5, 4), (5, 7), (6, 4), (6, 5), (7, 2), (8, 3), (8, 6)\}$$

1. Geben Sie für den Graphen eine Kantenliste an. (1 Punkt)
2. Geben Sie für den Graphen eine Knotenliste an. (1 Punkt)
3. Geben Sie für den Graphen eine Adjazenzliste an. (2 Punkte)
4. Erläutern Sie kurz den Algorithmus für den *Tiefendurchlauf* und überprüfen Sie mit diesem Algorithmus, ob der Graph zyklensfrei ist. (6 Punkte)