

Algorithmen und Datenstrukturen 1

Abschlussklausur WS 2011 / 2012

15. Februar 2012



Prof. Dr. Carsten Lecon
Fachhochschule Aalen
Fakultät Elektronik und Informatik
Studiengang Informatik

Name:	
Vorname:	
Semester:	
Matrikelnummer:	

- Bitte geben Sie diesen Zettel sowie die übrigen Aufgabenzettel zusammen mit den Lösungsblättern ab.
- Das erste Blatt der abgegebenen Klausur muss dieses **Deckblatt** sein.
- Die Blätter sind nach Aufgaben **aufsteigend sortiert**.
- Die Blätter sind **zusammengeheftet**. Nicht angeheftete Blätter werden nicht gewertet!

Viel Erfolg!

Einträge in der folgenden Tabelle sind nicht vom Prüfling auszufüllen
(dennoch getätigte Angaben werden ggf. korrigiert ...).

Aufgabe	Thema	Max. Punkte	Erreichte Punkte
1	Rekursion	12	
2	Queue/Liste	13	
3	RS-Baum	30	
4	BST	8	
5	Heasport	16	
6	Quicksort	10	
7	Sortieralg.	26	
8	Bubblesort	10	
9	Allg. Fragen	10	
Punkte aus Übungen			
SUMME		135	

Hinweise:

- 120 Punkte entsprechen 100%.
- Zum Bestehen der Klausur sind 60 Punkte erforderlich.

1. Rekursion [12]

Gegeben ist die folgende Funktion f (Wertebereich: alle natürlichen positiven Zahlen einschließlich 0):

$$f(x,y)=1 + f(x-y, y) \text{ für } x \geq y,$$

$$f(x,y)=0 \text{ sonst.}$$

a) Berechnen Sie die folgenden Funktionswerte: [4]:

(x, y)	(8,2)	(2,1)	(1,2)	(2,2)
$f(x, y)$				

b) Schreiben Sie eine Java-Methode, die den Funktionswert von $f(x,y)$ berechnet. Sie können davon ausgehen, dass nur positive natürliche Zahlen (einschließlich 0) übergeben werden. [2]

c) Stellen Sie den Aufruf von $f(8,2)$ grafisch dar. [4]

d) Welche mathematische Funktion wird hier beschrieben? [2]

LÖSUNG:

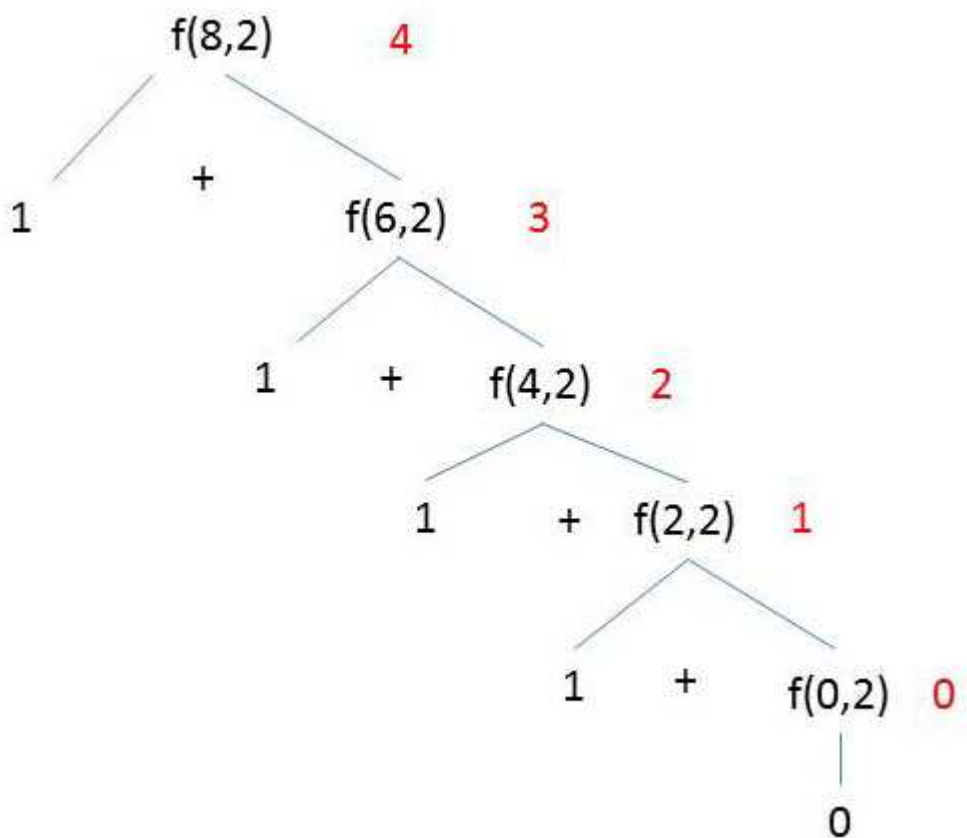
a)

(x, y)	(8,2)	(2,1)	(1,2)	(2,2)
f(x, y)	4	2	0	1

b)

```
int f(int x, int y) {  
    if (x >= y)  
        return 1 + f(x-y, y);  
    else  
        return 0;  
}
```

c)



d) Ganzzahlige Division (x/y).

2. Greedy-Algorithmus [13]

a) Realisieren Sie in Java einen Greedy-Algorithmus für das Herausgeben von Münzen. Folgende Münzwerte stehen zur Verfügung: 1, 2, 5, 10. Implementieren Sie die Methode `greedy` in der Klasse `Greedy`; die ermittelten Werte sollen auf dem Bildschirm ausgegeben werden:

```
public class Greedy {  
  
    private int[] werte = {1, 2, 5, 10};  
  
    public void greedy(int wert) {  
        ...  
    } // greedy  
}
```

Die ausgegebenen Werte sollen summiert exakt den Wert von `wert` besitzen.

Sie können davon ausgehen, dass die Werte im Array `werte` aufsteigend sortiert sind. [8]

Aufrufbeispiele:

Aufruf	Ausgabe
<code>greedy(101)</code>	10 10 10 10 10 10 10 10 10 10 1
<code>greedy(99)</code>	10 10 10 10 10 10 10 10 10 5 2 2
<code>greedy(17)</code>	10 5 2

b) Welche Werte werden beim Aufruf `greedy(19)` ausgegeben? [1]

c) Welche Laufzeit hat die Methode `greedy` im *worst case*? [1]

d) Was würde beim Aufruf von `greedy(17)` passieren, wenn das Array `werte` nur aus den Elementen {2, 5, 10} bestehen würde? Wie müssten Sie Ihr Programm ändern, wenn keine Lösung gefunden werden kann? [3]

LÖSUNG:

a)

```
public void greedy(int wert) {
    while (wert != 0) {
        for (int i=werte.length-1; i>=0; i--) {
            if (wert-werte[i] >= 0) {
                System.out.println(werte[i]);
                wert -= wert[i];
                break;
            } // if
        } // for
    } // while
} // greedy
```

b) 10 5 2 2

c) $O(n)$ (n: Anzahl der Werte)

d)

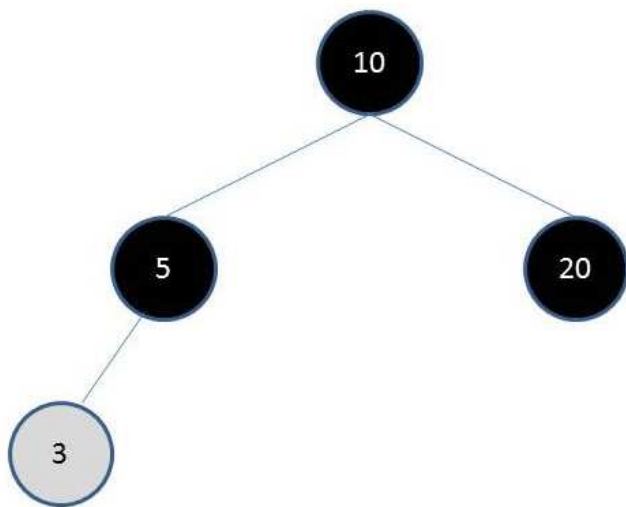
```
public void greedy(int wert) {
    while (wert != 0) {
        // Keine Loesung mehr moeglich?:
        if (wert - wert[0] < 0) {
            System.out.println("Keine Lösung möglich!");
            return;
        } // if
        for (int i=werte.length-1; i>=0; i--) {
            if (wert-werte[i] >= 0) {
                System.out.println(werte[i]);
                wert -= wert[i];
                break;
            } // if
        } // for
    } // while
} // greedy
```

3. Rot-Schwarz-Baum [30]

Fügen Sie in folgenden Rot-Schwarz-Baum (hellgraue Knoten sind rote Knoten) nacheinander die sechs Knoten mit den Schlüsseln

2
1
4
7
9
0

ein. Zeichnen Sie jeweils den resultierenden Baum. Geben Sie jeweils den Zustand direkt nach dem Einfügen und ggf. nach einer Wiederherstellung der Rot-Schwarz-Eigenschaft an. Notieren Sie dabei, welche Baum-Operationen Sie ggf. durchführen.



Bewertung: 1P für Anfügen an richtiger Position, 4P für eventuelle Aktion0



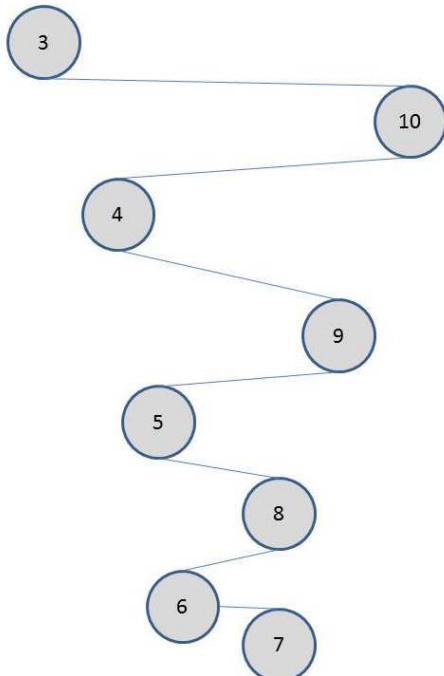


Aktionen:

- 2 RR(5)
- 1 1b
- 4 (keine)
- 7 (keine)
- 9 1b – LR(3) – RR(10)
- 0 RR(4)

4. Binäre Suchbäume [8]

- a) Was bedeutet „binär“ in „binärer Suchbaum“? [1]
- b) Fügen Sie die Zahlen 5, 3, 9, 6, 7, 1, 21, 2, 18 in dieser Reihenfolge in einen anfangs leeren binären Suchbaum ein. Zeichnen Sie den Suchbaum, nachdem Sie alle Zahlen eingefügt haben. [3]
- c) Löschen Sie aus dem Suchbaum, den Sie in der obigen Aufgabe erstellt haben, den Knoten mit dem Wert 9. Zeichnen Sie den resultierenden Suchbaum und begründen Sie Ihr Vorgehen. [2]
- d) Gegeben sei folgender binärer Suchbaum. In welcher Reihenfolge wurden die Zahlen in diesen Baum eingefügt? Gegeben Sie eine mögliche Lösung an. [2]



LÖSUNG:

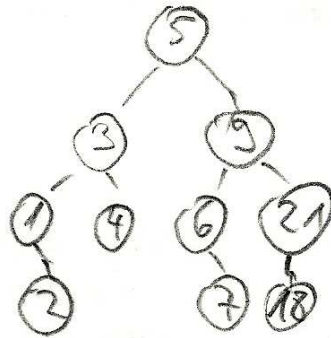
b), c):

10.2.2012

AL01: Warten

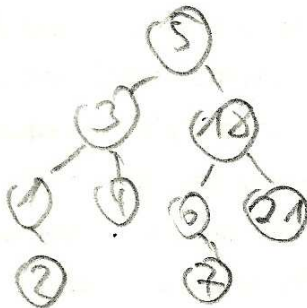
BST

5-3-⁹6-7-4-1-21-2-18



Löschen von 9:

Ersetzen von kleinstem Element in rechte Teilbaum
(18):



d) 3-10-4-9-5-8-6-7

5. Sortieren: Heapsort [16]

Gegeben sei die Zahlenfolge {3, 10, 4, 9, 5, 8, 6, 7}.

- Zeichnen Sie den entsprechenden Heap-Baum. [1]
- Handelt es sich dabei um einen Max- oder einen Min-Heap oder weder noch? Begründen Sie Ihre Antwort. [1]
- Sortieren Sie die obige Zahlenfolge mittels des Heapsort-Algorithmus. Tragen Sie die Zwischenstände in folgende Tabelle ein. [12]
- Zeichnen Sie den Heap-Baum nach der Sortierung. [1]
- Handelt es sich bei dem Heap-Baum nach der Sortierung um einen Max- oder einen Min-Heap oder weder noch? Begründen Sie Ihre Antwort. [1]

1	2	3	4	5	6	7	8	Array-Index
3	10	4	9	5	8	6	7	Ursprungs-Array
								Array nach Herstellung des Max-Heaps
								Array nach Sortierung
								Array nach Wiederherstellung des Max-Heaps
								Array nach Sortierung
								Array nach Wiederherstellung des Max-Heaps
								Array nach Sortierung
								Array nach Wiederherstellung des Max-Heaps
								Array nach Sortierung
								Array nach Wiederherstellung des Max-Heaps
								Array nach Sortierung
								Array nach Wiederherstellung des Max-Heaps
								Array nach Sortierung
								Array nach Wiederherstellung des Max-Heaps
								Array nach Sortierung
								Array nach Wiederherstellung des Max-Heaps
								Array nach endgültiger Sortierung

LÖSUNG:

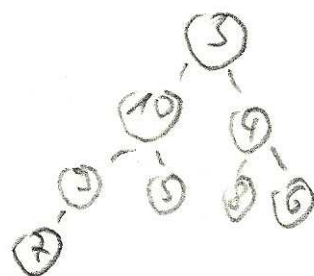
1	2	3	4	5	6	7	8	Array-Index
3	10	4	9	5	8	6	7	Ursprungs-Array
10	9	8	7	5	4	6	3	Array nach Herstellung des Max-Heaps
3	9	8	7	5	4	6	10	Array nach Sortierung
9	7	8	3	5	4	6		Array nach Wiederherstellung des Max-Heaps
6	7	8	3	5	4	9		Array nach Sortierung
8	7	6	3	5	4			Array nach Wiederherstellung des Max-Heaps
4	7	6	3	5	8			Array nach Sortierung
7	5	6	3	4				Array nach Wiederherstellung des Max-Heaps
4	5	6	3	7				Array nach Sortierung
6	5	4	3					Array nach Wiederherstellung des Max-Heaps
3	5	4	6					Array nach Sortierung
5	3	4						Array nach Wiederherstellung des Max-Heaps
4	3	5						Array nach Sortierung
4	3							Array nach Wiederherstellung des Max-Heaps
3	4							Array nach Sortierung
3								Array nach Wiederherstellung des Max-Heaps
3	4	5	6	7	8	9	10	Array nach endgültiger Sortierung

Algor - Heap

11.2.2012

Heap

0	1	2	3	4	5	6	7	8
3	10	4	2	5	8	6	7	



1.

3	10	4	2	5	8	6	7
---	----	---	---	---	---	---	---

3	10	8	2	5	4	6	7
---	----	---	---	---	---	---	---

10	3	8	2	5	4	6	7
----	---	---	---	---	---	---	---

10	9	8	3	5	4	6	7
----	---	---	---	---	---	---	---

10	9	8	7	5	4	6	3
----	---	---	---	---	---	---	---

13	2	8	7	5	4	6	10
----	---	---	---	---	---	---	----

2.

9	3	8	7	5	4	6
---	---	---	---	---	---	---

9	7	8	2	5	4	6
---	---	---	---	---	---	---

16	7	8	3	5	4	9
----	---	---	---	---	---	---

3.

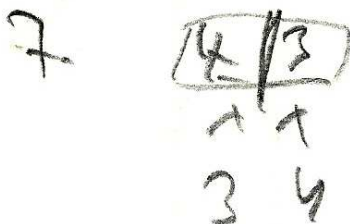
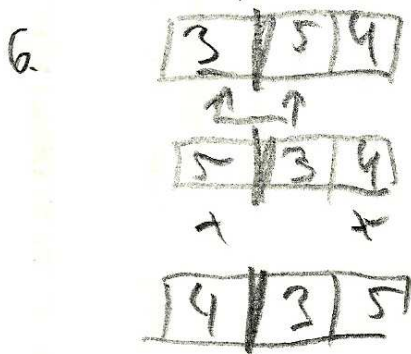
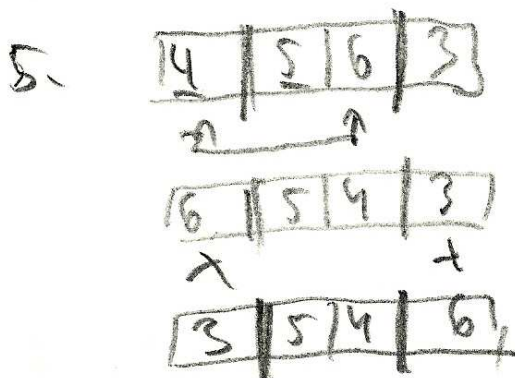
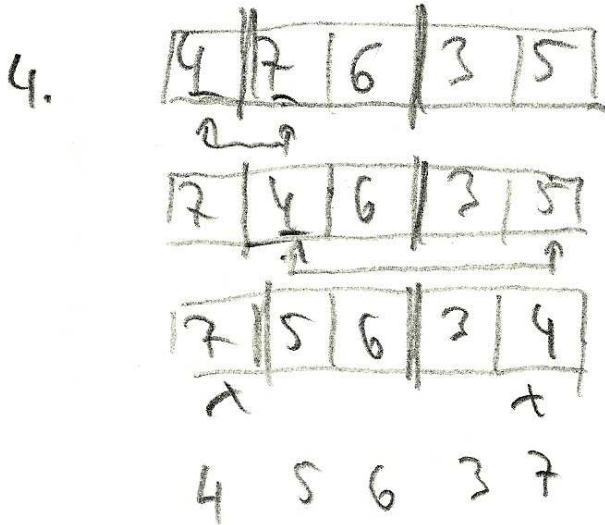
16	7	2	3	5	4
----	---	---	---	---	---

8	7	6	3	5	4
---	---	---	---	---	---

4 7 6 3 5 8

Alison - Wunsin
Kempal - Fudri

12.7.2012



6. Sortieren: Quicksort [10]

Gegeben sei die Zahlenfolge {3, 10, 4, 9, 5, 8, 6, 7}.

Sortieren Sie diese Zahlenfolge mit dem **Quicksort2 (!)**-Algorithmus aus der Vorlesung.
Geben Sie jeden Aufruf an.

Lösung 7. Quicksort2

QS2(1,8)

1	2	3	4	5	6	7	8
3	10	4	9	5	8	6	7
3	6	4	9	5	8	10	7
3	6	4	5	9	8	10	7
3	6	4	5	7	8	10	9

Pivot: 7, i=1, j=7

→ i=2, j=7 → swap(A[2],A[7])

→ i=4, j=5 → swap(A[4],A[5])

→ i=5, j=4 → (kein swap), A[5]>Pivot → swap

QS2(1,4) – QS(6,8)

QS2(1,4)

1	2	3	4
3	6	4	5
3	4	6	5

Pivot: 5, i=1, j=3

→ i= 2, j=3 → swap(A[2],A[3])

→ i=3, j=2 → kein swap, A[3]>Pivot → swap

QS2(1,2) – QS2(4,4)

QS2(1,2)

1	2
3	4

Pivot: 4, i=1, j=1

→ i=1, j=1 → kein swap, !A[1]>Pivot

QS2(1,1) – QS(2,2)

QS2(6,8)

1	2	3	4	5	6	7	8
					8	10	7
					7	10	8

Pivot: 7, i=6, j=7

→ i=6, j=6 → kein swap, A[6]>Pivot → swap

QS2(6,6) – QS2(7,8)

QS2(7,8)

1	2	3	4	5	6	7	8
						10	8
						8	10

Pivot: 8, i=7, j=7

→ i=7, j=7 → kein swap, A[7]>Pivot → swap

QS2(7,7) – QS2(8,8)

7.Sortieren: Mergesort, Insertionsort, Shellsort [26]

- a) Gegeben sei die Zahlenfolge {3, 10, 4, 9, 5, 8, 6, 7}.
- b) Sortieren Sie diese Zahlenfolge mit dem Mergesort-Algorithmus. Visualisieren Sie den Ablauf des Algorithmus durch Zeichnen des Rekursionsbaums. [8]
- c) Sortieren Sie diese Zahlenfolge mit dem Insertionsort-Algorithmus. Geben Sie die Zwischenschritte an. [8]
- d) Sortieren Sie diese Zahlenfolge mit dem Shellsort-Algorithmus. Geben Sie die Zwischenschritte an. [8]
- e) Was ist der Vorteil von Shellsort gegenüber Insertionsort? [2]

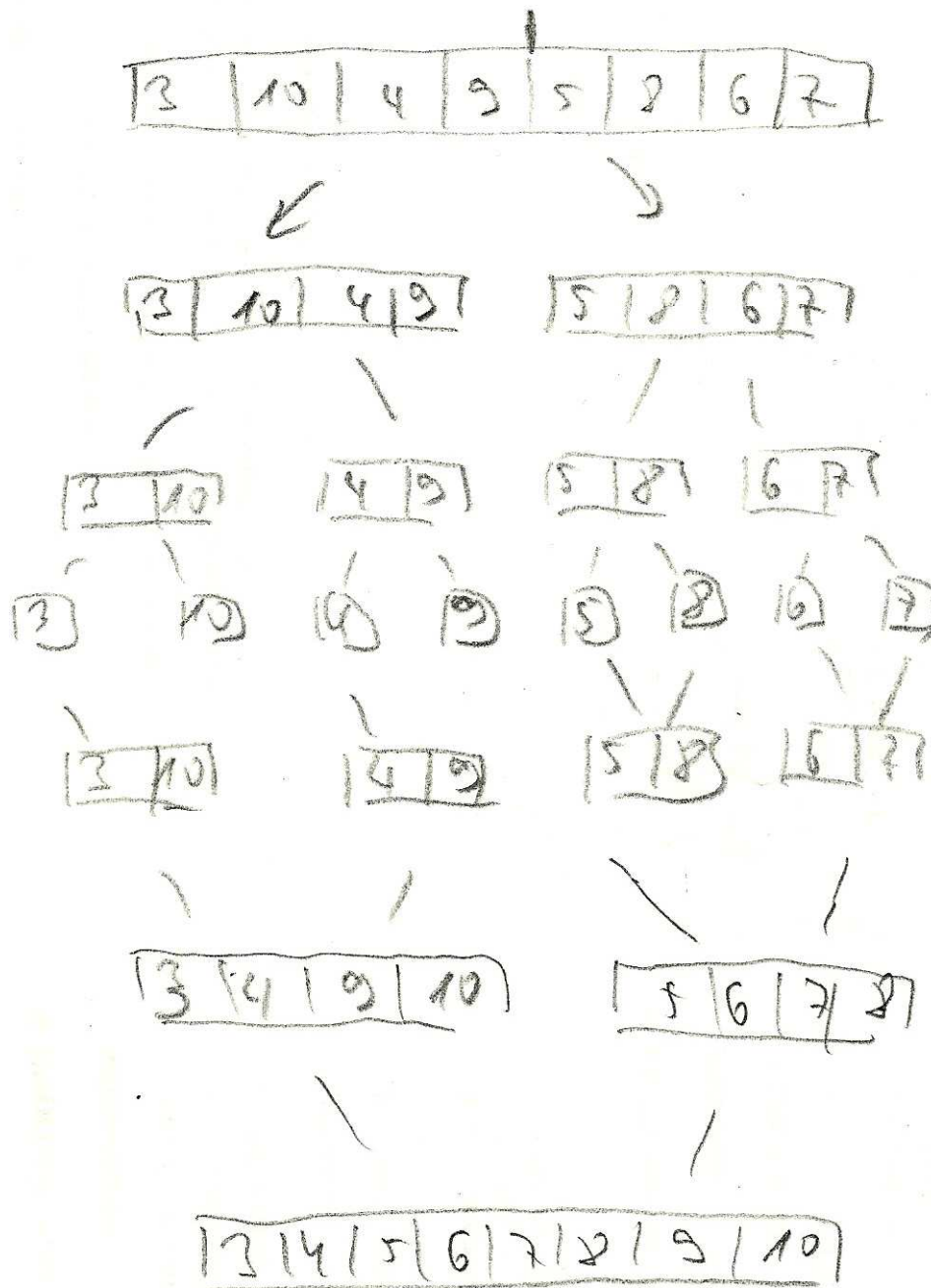
LÖSUNG:

a)

Algo1 - Mergesort

19.2.2012

Mergesort



Insertionsort

i	1	2	3	4	5	6	7	8
Orig	3	10	4	9	5	8	6	7
i=2	3	10	4	9	5	8	6	7
i=3	3	4	10	9	5	8	6	7
i=4	3	4	9	10	5	8	6	7
i=5	3	4	5	9	10	8	6	7
i=6	3	4	5	8	9	10	6	7
i=7	3	4	5	6	8	9	10	7
i=8	3	4	5	6	7	8	9	10

Shellsort

(Anfangs vier Teile)

i	1	2	3	4	5	6	7	8
Orig	3	10	4	9	5	8	6	7
	(i)	(ii)	(iii)	(iv)	(i)	(ii)	(iii)	(iv)
	3	8	4	7	5	10	6	9
	(i)	(ii)	(i)	(ii)	(i)	(ii)	(i)	(ii)
	3	7	4	8	5	9	6	10
i=2	3	7	4	8	5	9	6	10
i=3	3	4	7	8	5	9	6	10
i=4	3	4	7	8	5	9	6	10
i=5	3	4	5	7	8	9	6	10
i=6	3	4	5	7	8	9	6	10
i=7	3	4	5	6	7	8	9	10
i=8	3	4	5	6	7	8	9	10

Vorteil von Shellsort: Vorsortierung, dadurch werden weit entfernte Zahlen in die Nähe der richtigen Position gebracht.

8. Sortieren: Bubblesort [10]

Gegeben sei die Zahlenfolge {3, 10, 4, 9, 5, 8, 6, 7}.

- a) Sortieren Sie diese Zahlenfolge mit dem Bubblesort-Algorithmus (2. Variante).
Schreiben Sie die Zahlenfolge nach jedem Durchlauf der äußeren Schleife in nachfolgende Tabelle
(die Anzahl der Zeilen der Tabelle spiegelt nicht unbedingt die tatsächliche Anzahl von
Schleifendurchläufen wider...). [8]
- b) Was ist der Vorteil der zweiten Variante von Bubblesort gegenüber der ersten? [2]

Durchlauf								
0	3	10	4	9	5	8	6	7
1								
2								

LÖSUNG:

Durchlauf								
0	3	10	4	9	5	8	6	7
1	3	4	9	5	8	6	7	10
2	3	4	5	8	6	7	9	10
3	3	4	5	6	7	8	9	10
4	3	4	5	6	7	8	9	10

9. Allgemeine Fragen [10]

- a. Welche der folgenden Laufzeiten bezeichnet den schnellsten, den zweitschnellsten und den drittschnellsten Algorithmus? [1]

- a. exponentielle Laufzeit
- b. kubische Laufzeit
- c. Laufzeit von $n \log(n)$

Schnellste Laufzeit: ____

Zweitschnellste Laufzeit: ____

Drittschnellste Laufzeit: ____

- b. Was ist der Vorteil von binären Suchbäumen gegenüber „normalen“ Binärbäumen? [2]

- c. Erläutern Sie das Prinzip der binären Suche in einem Array. Welche Voraussetzungen müssen erfüllt sein? [2]

- d. Sie müssen zu einer Geschäftsbesprechung, dürfen aber nur einen 20 cm hohen Aktenkoffer mitnehmen. Darin möchten Sie so viele Aktenordner wie möglich stecken. Die Dicke der Aktenordner sieht wie folgt aus:

Akte	Höhe
Vorlesungsskripte	5cm
Anleitung Super Mario	4cm
Algorithmen	6cm
Kochbuch für Profis	4cm
DIN ISO 9000	15cm
1000 neue Witze	2cm
Telefonbuch	4cm

Packen Sie nun den Aktenordner mit Akten. Verwenden Sie dabei den Greedy-Algorithmus. [2]

Akte	Einpacken?
Vorlesungsskripte	
Anleitung Super Mario	
Algorithmen	
Kochbuch für Profis	
DIN ISO 9000	
1000 neue Witze	
Telefonbuch	

Warum wird mit diesem Algorithmus nicht die maximal mögliche Anzahl von Aktenordner erreicht?
Wie sieht die optimale Lösung aus, wenn Sie nicht den Greedy-Algorithmus verwenden? [3]

LÖSUNG:

Welche der folgenden Laufzeiten bezeichnet den schnellsten, den zweitschnellsten und den drittschnellsten Algorithmus? [1]

- a. exponentielle Laufzeit
- b. kubische Laufzeit
- c. Laufzeit von $n \log(n)$

Schnellste Laufzeit: c

Zweitschnellste Laufzeit: b

Drittschnellste Laufzeit: a

b.

Was ist der Vorteil von binären Suchbäumen gegenüber „normalen“ Binärbäumen? [2]

Effizienteres Suchen, da die Werte sortiert abgelegt sind.

- c. Erläutern Sie das Prinzip der binären Suche in einem Array. Welche Voraussetzungen müssen erfüllt sein? [2]

In der Mitte beginnen und dann je nach Relation links oder rechts rekursiv weitersuchen.

Voraussetzung ist, dass die Werte im Array sortiert sind.

- d. Sie müssen zu einer Geschäftsbesprechung, dürfen aber nur einen 20 cm hohen Aktenkoffer mitnehmen. Darin möchten Sie so viele Aktenordner wie möglich stecken. Die Dicke der Aktenordner sieht wie folgt aus::

Akte	Höhe
Vorlesungsskripte	5cm
Anleitung Super Mario	4cm
Algorithmen	6cm
Kochbuch für Profis	4cm
DIN ISO 9000	15cm
1000 neue Witze	2cm
Telefonbuch	4cm

Packen Sie nun den Aktenordner mit Akten. Verwenden Sie dabei den Greedy-Algorithmus. [2]

Akte	Einpacken?	Besser
Vorlesungsskripte	X	
Anleitung Super Mario		
Algorithmen		
Kochbuch für Profis		
DIN ISO 9000	X	
1000 neue Witze		
Telefonbuch		

Warum wird mit diesem Algorithmus nicht die maximal mögliche Anzahl von Aktenordner erreicht? Wie sieht die optimale Lösung aus, wenn Sie nicht den Greedy-Algorithmus verwenden? [3]

1000 neue Witze (2), Anleitung Super Mario (+4 = 6), Kochbuch für Profis (+4=10), Telefonbuch (+4 = 14), Algorithmen (+6=20).