

## Übung 12 zum Modul

### Algorithmen und Datenstrukturen im WS 2005/2006

**Prof. Dr. W. Kowalk,**

Abteilung Rechnernetze und Telekommunikation  
Department Informatik, Fakultät II, Universität Oldenburg

<b>12</b>	Dieses Übungsblatt ist nicht obligatorisch. Es kann zur Verbesserung verwendet werden.		
<b>12.1</b>	<b>Allgemeine Fragen (20 Punkte)</b>		
a)	Hinweis zu der folgenden Tabelle: Richtige Antworten werden mit zwei Pluspunkten, falsche mit zwei Minuspunkten bewertet. Wenn die Summe aller Punkte negativ ist, wird diese Teilaufgabe mit null Punkten bewertet. <b>(20 Punkte)</b>		
<b>Nr</b>	<b>Behauptung</b>	<b>stimmt</b>	<b>stimmt nicht</b>
1	Informationsgehalt und Entropie haben stets den gleichen Wert.		
2	Die ideelle Entropie ist größer oder gleich der wahren Entropie.		
3	Die Formel für die wahre Entropie $H$ lautet $H = \sum_i p_i \cdot \log_{10} p_i$ .		
4	Bubblesort ist der langsamste direkte Sortieralgorithmus.		
5	Sortieren durch Einfügen wird durch gleichzeitiges Einfügen mehrerer Elemente deutlich verbessert.		
6	Das Halteproblem kann nur von einer Turingmaschine gelöst werden.		
7	Ein Codebaum für $N$ Zustände hat stets die Tiefe $\log_2(N)$ .		
8	Polymorphie kann nur mit objektorientierten Sprachen realisiert werden.		
9	Im 3AA lässt sich nicht die fünfte Wurzel aus einer Gleitpunktzahl ziehen.		
10	Ein indeterminierter Algorithmus terminiert nie.		
<b>12.1</b>	<b>Allgemeine Fragen (30 Punkte)</b>		
b)	i) Vergleichen Sie die Begriffe Überladung und Überschreiben. ii) Nennen Sie fünf Eigenschaften, die Algorithmen zukommen können; erläutern Sie jede Eigenschaft kurz. iii) Erläutern Sie die Begriffe <i>Laufzeitkomplexität</i> , <i>Komplexitätsparameter</i> , <i>O-Notation</i> . Wie wird <i>quadratische</i> , wie <i>exponentielle Laufzeit</i> notiert? Was bedeutet das?		
<b>12.2</b>	<b>3AA Programmierung (50 Punkte)</b>		
	Lösen Sie <u>zwei</u> der folgenden Probleme durch ein 3AA-Programm.		
a)	Ziehen Sie die siebte Wurzel durch Intervallhalbierung. Verwenden Sie keine der eingebauten analytischen Funktionen (Radizieren, Potenzieren usw.) von 3AA!		
b)	Analysieren Sie das Programm 'Tannenbaum'. Ändern Sie es so um, dass es Frohe Ostern wünscht und einen Baum mit runder Krone (oder ein Osterei) zeichnet.		
c)	Kopieren Sie einen Text; ändern Sie die Zeichenfolgen 'ae', 'oe', 'ue', 'ss' in 'ä', 'ö', 'ü', 'ß' um (unabhängig von neuer oder alter Rechtschreibung!).		

Hinweise: Die Übungen sind bis Mittwoch, 1.2.2006, 10.00 Uhr in den Kästen der jeweiligen Tutoren abzugeben.

```

1      // Tannenbaum
2
3      adr setColor := con 5 // Stamm, Farbe
4      adr PLOTTER := con 6 // zeichne Hintergrundfarbe mit setColor
5      adr PLOTTER := con 0 // lösche Zeichenfläche
6      adr setSize := con 160 // Stamm, Dicke
7      adr PLOTTER := con 4 // übernehme Größe von isSize
8      adr setColor := con 4 // Stamm, Farbe
9      adr PLOTTER := val x3, con 600 // zeichne von (x3,600)
10     adr PLOTTER := val x3, con 500 // nach (x3,500)
11     adr setSize := con 20 // Blatt, Dicke
12     adr PLOTTER := con 4 // übernehme Größe von isSize
13 tannenbaum adr setColor := con 7 // setze Größe+Farbe
14     adr PLOTTER := val x1, val y1 // zeichne nach (x1,y1)
15     adr PLOTTER := val x2, val y1 // zeichne nach (x2,y1)
16     adr PLOTTER := val x3, val y2 // zeichne nach (x3,y2)
17     adr PLOTTER := val x1, val y1 // zeichne nach (x1,y1)
18     adr x1 += con 5 // neues x1
19     adr x2 -= con 5 // neues x2
20     adr y1 := val y2 - con 1 // neues y1
21     adr y2 := val y2 - con 5 // neues y2
22     if val x1 < val x2 then goto con tannenbaum // wenn noch nicht an
der Spitze, noch mal
23     adr setSize := con 2 * val groesse // Textgröße
24     adr PLOTTER := con 4 // übernehme Textgröße
25     adr setColor := con 8 // Schriftfarbe
26     adr PLOTTER := con 5 // übernehme Schriftfarbe
27     adr setX := con 210 // setze Anfangskoordinate Schrift
28     adr setY := con 200
29     jsr con textOutput // Gebe Schrift aus
30     stop
31 x1 con 100
32 x2 con 700
33 x3 con 400
34 y1 con 500
35 y2 con 495
36 groesse con 20
37
38
39
40     //////////////// Textausgabe ////////////////
41 textStart con Text1
42 textEnd con Text2
43 con 0
44 textOutput if val textStart >= con Text2 then goto con Ende
45     adr setChar := val val textStart // Zeichen nach Plotter
46     adr PLOTTER := con 3 // gebe Zeichen aus
47     adr setX := val setX + val groesse // setze neue Adresse
48     adr setY := val setY - con 1 // auch y wird verändert
49     adr textStart := val textStart + con 1 // zeiger auf neues Zeichen
50     goto con textOutput // noch mal
51 Ende rts con textOutput // zurück zum Aufruf
52
53 PLOTTER // Plotter-Eingabewerte
54 setX con 50 // adr PLOTTER := 2
55 setY con 50
56 setChar con 'A' // adr PLOTTER := 3
57 setSize con 20 // adr PLOTTER := 4
58 setColor con 3 // adr PLOTTER := 5
59 Text1 con "Frohe Weihnachten!"
60 Text2

```