Zusätzliche Übungsaufgaben (Lösungen): Algorithmen und Datenstrukturen

Fakultät für Fahrzeugtechnik
Ostfalia – Hochschule für angewandte Wissenschaften

(Hinweis: Sie sollten die folgenden Aufgaben in 60 Minuten vollständig bearbeiten können)

Wichtige Hinweise für die Klausur:

- Es sind nur die folgenden Hilfsmittel erlaubt:
  - Nichtprogrammierbarer Taschenrechner
  - Tintenstifte (Kugelschreiber, Füller, Filzstifte o.ä.) in blau und/und schwarz
  - Lineal und/oder Geodreieck
- Schalten Sie alle anderen technischen Geräte vor Klausurbeginn aus und verstauen Sie diese sowie alle anderen nicht zugelassenen Gegenstände in Ihrer Tasche. Verschließen Sie bitte Ihre Tasche.
- Es gelten die Bestimmungen Ihrer Prüfungsordnung, insbesondere hinsichtlich Versäumnis, Täuschungsversuch und Ordnungsverstoß.
- Bearbeiten Sie alle Aufgaben einschließlich aller Nebenrechnungen auf den ausgeteilten Aufgabenblättern. Eigenes Papier ist nicht zugelassen. Kontrollieren Sie die ausgegebenen Bögen auf Vollständigkeit.
- Schreiben Sie auf jedes Blatt in der dafür vorgesehenen Kopfzeile Ihren Vor- und Nachnamen sowie Ihre Matrikelnummer.

Nachname, Vorname: ______________________________ Matrikelnummer: _________________

Aufgabe 1: Multiple Choice

Bewerten Sie die die folgenden Aussagen mit „richtig“ oder „falsch“ durch Ankreuzen in der jeweiligen Tabellenspalte. Eine richtige Bewertung wird mit +1 Punkt gewertet, eine falsche mit -1 Punkt. Fehlt eine Bewertung, wird dies mit 0 Punkten gewertet. Bei einer negativen Gesamtpunktzahl wird diese Aufgabe mit 0 Punkten gewertet.

	richtig	falsch
Es gibt Algorithmen, deren Beschreibung unendlich lang ist.		X
Der abstrakte Datentyp *Kellerspeicher* arbeitet nach dem LIFO-Prinzip.	X	
Der abstrakte Datentyp *Warteschlange* stellt eine Funktion *pop* bereit.		X
Der Datentyp *Verkettete Liste* kann rekursiv definiert werden.	X	
Der Algorithmus binäre Suche arbeitet nach dem Teile-und-Herrsche-Prinzip.	X	
Hash-Funktionen sind i. A. injektiv.		X
Bei geschlossenem Hashing wird bei Kollisionen eine feste Sprungfolge durchlaufen.	X	
Der abstrakte Datentyp *Kellerspeicher* arbeitet nach dem LIFO-Prinzip.	X	
Die Worst-Case-Laufzeit von Quicksort liegt in $\mathcal{O}(n^2)$.	X	
Die Worst-Case-Laufzeit von Heap-Sort liegt in $\mathcal{O}(n \log(n))$.	X	

(/10 Punkte)

Aufgabe 2: Turing-Maschine

Gegeben ist eine Turing-Maschine $T = (Z, \Sigma, \Gamma, \delta, z_0, \cdot, E)$ mit

- $Z = \{z_0, z_1, z_2, z_3, z_e\}$ (Zustandsmenge)
- $\Sigma = \{|\}$ (Eingabealphabet)
- $\Gamma = \{\cdot, |, h\}$ (Bandalphabet)
- $E = \{z_e\}$ (Endzustände)

Nachname, Vorname: ______________________________ Matrikelnummer: ________________

- δ: (Überführungsfunktion)

Zustand	liest	Aktion	Folgezustand
z_0	$\cdot$	L	z_2
z_0	$\vert$	L	z_0
z_0	h	L	z_2
z_1	h	R	z_1
z_1	$\vert$	R	z_1
z_1	$\cdot$	$\vert$	z_0
z_2	h	L	z_2
z_2	$\vert$	h	z_1
z_2	$\cdot$	R	z_3
z_3	h	$\vert$	z_3
z_3	$\vert$	R	z_3
z_3	$\cdot$	S	z_e

Bitte beachten Sie die Konvention, dass zu Beginn und zum Ende der Berechnung der Lese-/Schreibkopf auf dem ersten Leerzeichen rechts neben dem Ein- bzw. Ausgabewort steht!

a) Geben Sie für das Eingabewort

$$||||$$

die Zustandsfolge und die Bandkonfigurationen an, die T durchläuft. Markieren Sie die jeweilige Position des Lese-Schreibkopfes mit einem ^ unter der entsprechenden Bandposition.

Zustand	Bandkonfiguration
z_0	$\cdots\cdots\,||||\,\cdots\cdots$ $\wedge$
z_2	$\cdots\cdots\,||||\,\cdots\cdots$ $\wedge$
z_1	$\cdots\cdots\,|||h\,\cdots\cdots$ $\wedge$
z_1	$\cdots\cdots\,|||h\,\cdots\cdots$ $\wedge$
z_0	$\cdots\cdots\,|||h|\,\cdots\cdots$ $\wedge$
z_0	$\cdots\cdots\,|||h|\,\cdots\cdots$ $\wedge$
z_2	$\cdots\cdots\,|||h|\,\cdots\cdots$ $\wedge$
z_1	$\cdots\cdots\,||hh|\,\cdots\cdots$ $\wedge$
z_1	$\cdots\cdots\,||hh|\,\cdots\cdots$ $\wedge$
z_1	$\cdots\cdots\,||hh|\,\cdots\cdots$ $\wedge$
z_1	$\cdots\cdots\,||hh|\,\cdots\cdots$ $\wedge$
z_0	$\cdots\cdots\,||hh||\,\cdots\cdots$ $\wedge$

Nachname, Vorname: ______________________________ Matrikelnummer: __________________

z_0	$\cdots \; \| \; \| \; h \; h \; \underset{\wedge}{\|} \; \| \; \cdots$
z_0	$\cdots \; \| \; \| \; h \; \underset{\wedge}{h} \; \| \; \| \; \cdots$
z_2	$\cdots \; \| \; \| \; \underset{\wedge}{h} \; h \; \| \; \| \; \cdots$
z_2	$\cdots \; \| \; \underset{\wedge}{\|} \; h \; h \; \| \; \| \; \cdots$
z_1	$\cdots \; \| \; \underset{\wedge}{h} \; h \; h \; \| \; \| \; \cdots$
z_1	$\cdots \; \| \; h \; \underset{\wedge}{h} \; h \; \| \; \| \; \cdots$
z_1	$\cdots \; \| \; h \; h \; \underset{\wedge}{h} \; \| \; \| \; \cdots$
z_1	$\cdots \; \| \; h \; h \; h \; \underset{\wedge}{\|} \; \| \; \cdots$
z_1	$\cdots \; \| \; h \; h \; h \; \| \; \underset{\wedge}{\|} \; \cdots$
z_1	$\cdots \; \| \; h \; h \; h \; \| \; \| \; \underset{\wedge}{\;} \cdots$
z_0	$\cdots \; \| \; h \; h \; h \; \| \; \| \; \underset{\wedge}{\|} \; \cdots$
z_0	$\cdots \; \| \; h \; h \; h \; \| \; \underset{\wedge}{\|} \; \| \; \cdots$
z_0	$\cdots \; \| \; h \; h \; h \; \underset{\wedge}{\|} \; \| \; \| \; \cdots$
z_0	$\cdots \; \| \; h \; h \; \underset{\wedge}{h} \; \| \; \| \; \| \; \cdots$
z_2	$\cdots \; \| \; h \; \underset{\wedge}{h} \; h \; \| \; \| \; \| \; \cdots$
z_2	$\cdots \; \| \; \underset{\wedge}{h} \; h \; h \; \| \; \| \; \| \; \cdots$
z_2	$\cdots \; \underset{\wedge}{\|} \; h \; h \; h \; \| \; \| \; \| \; \cdots$
z_1	$\cdots \; \underset{\wedge}{h} \; h \; h \; h \; \| \; \| \; \| \; \cdots$
z_1	$\cdots \; h \; \underset{\wedge}{h} \; h \; h \; \| \; \| \; \| \; \cdots$
z_1	$\cdots \; h \; h \; \underset{\wedge}{h} \; h \; \| \; \| \; \| \; \cdots$
z_1	$\cdots \; h \; h \; h \; \underset{\wedge}{h} \; \| \; \| \; \| \; \cdots$
z_1	$\cdots \; h \; h \; h \; h \; \underset{\wedge}{\|} \; \| \; \| \; \cdots$
z_1	$\cdots \; h \; h \; h \; h \; \| \; \underset{\wedge}{\|} \; \| \; \cdots$
z_1	$\cdots \; h \; h \; h \; h \; \| \; \| \; \underset{\wedge}{\|} \; \cdots$
z_1	$\cdots \; h \; h \; h \; h \; \| \; \| \; \| \; \underset{\wedge}{\;} \cdots$

z_0	$\cdots\, h\, h\, h\, h \vert \vert \vert \underset{\wedge}{\vert} \,\cdots$
z_0	$\cdots\, h\, h\, h\, h \vert \vert \underset{\wedge}{\vert} \vert \,\cdots$
z_0	$\cdots\, h\, h\, h\, h \vert \underset{\wedge}{\vert} \vert \vert \,\cdots$
z_0	$\cdots\, h\, h\, h\, h \underset{\wedge}{\vert} \vert \vert \vert \,\cdots$
z_0	$\cdots\, h\, h\, h\, \underset{\wedge}{h} \vert \vert \vert \vert \,\cdots$
z_2	$\cdots\, h\, h\, \underset{\wedge}{h}\, h \vert \vert \vert \vert \,\cdots$
z_2	$\cdots\, h\, \underset{\wedge}{h}\, h\, h \vert \vert \vert \vert \,\cdots$
z_2	$\cdots\, \underset{\wedge}{h}\, h\, h\, h \vert \vert \vert \vert \,\cdots$
z_2	$\cdots \underset{\wedge}{\cdot}\, h\, h\, h\, h \vert \vert \vert \vert \,\cdots$
z_3	$\cdots\, \underset{\wedge}{h}\, h\, h\, h \vert \vert \vert \vert \,\cdots$
z_3	$\cdots\, \underset{\wedge}{\vert}\, h\, h\, h \vert \vert \vert \vert \,\cdots$
z_3	$\cdots\, \vert\, \underset{\wedge}{h}\, h\, h \vert \vert \vert \vert \,\cdots$
z_3	$\cdots\, \vert \underset{\wedge}{\vert}\, h\, h \vert \vert \vert \vert \,\cdots$
z_3	$\cdots\, \vert \vert\, \underset{\wedge}{h}\, h \vert \vert \vert \vert \,\cdots$
z_3	$\cdots\, \vert \vert \underset{\wedge}{\vert}\, h \vert \vert \vert \vert \,\cdots$
z_3	$\cdots\, \vert \vert \vert\, \underset{\wedge}{h} \vert \vert \vert \vert \,\cdots$
z_3	$\cdots\, \vert \vert \vert \underset{\wedge}{\vert} \vert \vert \vert \vert \,\cdots$
z_3	$\cdots\, \vert \vert \vert \vert \underset{\wedge}{\vert} \vert \vert \vert \,\cdots$
z_3	$\cdots\, \vert \vert \vert \vert \vert \underset{\wedge}{\vert} \vert \vert \,\cdots$
z_3	$\cdots\, \vert \vert \vert \vert \vert \vert \underset{\wedge}{\vert} \vert \,\cdots$
z_3	$\cdots\, \vert \vert \vert \vert \vert \vert \vert \underset{\wedge}{\vert} \,\cdots$
z_3	$\cdots\, \vert \vert \vert \vert \vert \vert \vert \vert \underset{\wedge}{\cdot} \cdots$
z_e	$\cdots\, \vert \vert \vert \vert \vert \vert \vert \vert \underset{\wedge}{\cdot} \cdots$

(/8 Punkte)

b) Was berechnet T?
Multiplikation mit 2 (auch richtig: T verdoppelt die Anzahl der Striche auf dem Band.)

(/2 Punkte)

Nachname, Vorname: ________________________________ Matrikelnummer: __________________

Aufgabe 3: Komplexitätsklassen

Untersuchen Sie, ob für folgende Funktionen gilt:

- $f(n) \in \mathcal{O}(g(n))$
- $f(n) \in \Omega\big(g(n)\big)$
- $f(n) \in \Theta(g(n))$

Es gelten dabei die Definitionen (mit $n, n_0 \in \mathbb{N}$ und $c, c_1, c_2 \in \mathbb{R}^+$)

- $\mathcal{O}\big(g(n)\big) = \{f(n) \mid \exists\, c > 0, n_0 \in \mathbb{N}, \forall\, n > n_0 : 0 \leq f(n) \leq c \cdot g(n)\}$
- $\Omega\big(g(n)\big) = \{f(n) \mid \exists\, c > 0, n_0 \in \mathbb{N}, \forall\, n > n_0 : 0 \leq c \cdot g(n) \leq f(n)\}$
- $\Theta\big(g(n)\big) = \{f(n) \mid \exists\, c_1, c_2 > 0, n_0 \in \mathbb{N}, \forall\, n > n_0 : 0 \leq c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)\}$

Außerdem dürfen Sie als bekannt voraussetzen:

- $f \in \Theta(g) \Leftrightarrow f \in \mathcal{O}(g) \wedge f \in \Omega(g)$
- $f \in \mathcal{O}(g) \Leftrightarrow g \in \Omega(f)$

Nachname, Vorname: ______________________________ Matrikelnummer: _________________

a) $f(n) = 10^{-6}n^3 + 25, \quad g(n) = n^3$

Behauptung: $f(n) \in \Theta\big(g(n)\big)$:

Beweis: $f(n) \in \mathcal{O}\big(g(n)\big)$:

$f(n) \in \mathcal{O}\big(g(n)\big) \iff \exists\, c > 0, n_0 \in \mathbb{N}, \forall\, n > n_0 : 0 \leq f(n) \leq c \cdot g(n)$ (mit $n \in \mathbb{N}$ und $c \in \mathbb{R}^+$)

$0 \leq 10^{-6}n^3 + 25 \leq c \cdot n^3$

$0 \leq 10^{-6} + \dfrac{25}{n^3} \leq c$

Wähle $n_0 = 3$ und $c = 1$:

$0 \leq 10^{-6} + \frac{25}{27} \leq 1$ ✓

Zeige, dass die Behauptung auch für alle $n > n_0$ gilt:

$0 \leq 10^{-6} + \frac{25}{n^3} \leq 1$ ✓

geht für große n gegen null

Beweis: $f(n) \in \Omega\big(g(n)\big)$:

$f(n) \in \Omega\big(g(n)\big) \iff \exists\, c > 0, n_0 \in \mathbb{N}, \forall\, n > n_0 : 0 \leq c \cdot g(n) \leq f(n)$ (mit $n \in \mathbb{N}$ und $c \in \mathbb{R}^+$)

$0 \leq c \cdot n^3 \leq 10^{-6}n^3 + 25$

$0 \leq c \leq 10^{-6} + \dfrac{25}{n^3}$

Wähle $n_0 = 1$ und $c = 10^{-7}$:

$0 \leq 10^{-7} \leq 10^{-6} + \frac{25}{1}$ ✓

Zeige, dass die Behauptung auch für alle $n > n_0$ gilt:

$0 \leq 10^{-7} \leq 10^{-6} + \frac{25}{n^3}$ ✓

auch für große n stets größer als 10^{-7}

□

Mit $f \in \mathcal{O}(g(n)) \wedge f \in \Omega(g(n)) \Rightarrow f(n) \in \Theta\big(g(n)\big)$

□

(/6 Punkte)

Nachname, Vorname: ________________________________ Matrikelnummer: __________________

a) $f(n) = 7n \ln n, \quad g(n) = 35n + 9$

Behauptung: $f(n) \in \Omega\big(g(n)\big)$:

Beweis:

$f(n) \in \Omega\big(g(n)\big) \iff \exists\, c > 0, n_0 \in \mathbb{N}, \forall\, n > n_0 : 0 \le c \cdot g(n) \le f(n)$ (mit $n \in \mathbb{N}$ und $c \in \mathbb{R}^+$)

$0 \le c \cdot (35n + 9) \le 7n \cdot \ln n$
$0 \le c \cdot \left(5 + \frac{9}{7n}\right) \le \ln n$
Wähle $n_0 = 5000$ und $c = 1$:
$0 \le \left(5 + \frac{9}{35000}\right) \le 8{,}517 \checkmark$
Zeige, dass die Behauptung auch für alle $n > n_0$ gilt:
$0 \le \left(5 + \frac{9}{7n}\right) \le \ln n \checkmark$

geht für große n gegen ∞

geht für große n gegen null

$\square$

Zeige: $f(n) \notin \mathcal{O}\big(g(n)\big)$

$f(n) \in \mathcal{O}\big(g(n)\big) \iff \exists\, c > 0, n_0 \in \mathbb{N}, \forall\, n > n_0 : 0 \le f(n) \le c \cdot g(n)$ (mit $n \in \mathbb{N}$ und $c \in \mathbb{R}^+$)

$0 \le 7n \cdot \ln n \le c \cdot (35n + 9)$
$0 \le \ln n \le c \cdot (5 + \frac{9}{7n})$ $\rightarrow$ so ein c kann es nicht geben: $\forall c \exists n\!: \ln n > c \cdot (5 + \frac{9}{7n})$

$f(n) \notin \mathcal{O}\big(g(n)\big) \Rightarrow f(n) \notin \Theta\big(g(n)\big)$

$\square$

(/6 Punkte)

Nachname, Vorname: ______________________________ Matrikelnummer: ________________

Aufgabe 4: Sortieren

Gegeben ist das folgende Array A:

-4	**3**	**2**	**2**	**7**	**3**	**5**	**7**	**0**	**1**

a) Sortieren Sie A aufsteigend mit Hilfe des Algorithmus *Bubblesort*. Geben Sie den Ablauf des Algorithmus schrittweise und vollständig in tabellarischer Form an. Schritte, in denen sich das Array nicht ändert, schreiben Sie bitte nicht mit auf.

-4	**3**	**2**	**2**	**7**	**3**	**5**	**7**	**0**	**1**
-4	**2**	**3**	**2**	**7**	**3**	**5**	**7**	**0**	**1**
-4	**2**	**2**	**3**	**7**	**3**	**5**	**7**	**0**	**1**
-4	**2**	**2**	**3**	**3**	**7**	**5**	**7**	**0**	**1**
-4	**2**	**2**	**3**	**3**	**5**	**7**	**7**	**0**	**1**
-4	**2**	**2**	**3**	**3**	**5**	**7**	**0**	**7**	**1**
-4	**2**	**2**	**3**	**3**	**5**	**7**	**0**	**1**	**7**
-4	**2**	**2**	**3**	**3**	**5**	**0**	**7**	**1**	**7**
-4	**2**	**2**	**3**	**3**	**5**	**0**	**1**	**7**	**7**
-4	**2**	**2**	**3**	**3**	**0**	**5**	**1**	**7**	**7**
-4	**2**	**2**	**3**	**3**	**0**	**1**	**5**	**7**	**7**
-4	**2**	**2**	**3**	**0**	**3**	**1**	**5**	**7**	**7**
-4	**2**	**2**	**3**	**0**	**1**	**3**	**5**	**7**	**7**
-4	**2**	**2**	**0**	**3**	**1**	**3**	**5**	**7**	**7**
-4	**2**	**2**	**0**	**1**	**3**	**3**	**5**	**7**	**7**
-4	**2**	**0**	**2**	**1**	**3**	**3**	**5**	**7**	**7**
-4	**2**	**0**	**1**	**2**	**3**	**3**	**5**	**7**	**7**
-4	**0**	**2**	**1**	**2**	**3**	**3**	**5**	**7**	**7**
-4	**0**	**1**	**2**	**2**	**3**	**3**	**5**	**7**	**7**

(/6 Punkte)

b) Arbeitet *Bubblesort* „in place“? Begründen Sie Ihre Antwort.

Ja, Bubblesort arbeitet „in place“, weil die Sortierung allein durch Vertauschung von Einträgen im Eingabearray erfolgt. Es wird (abgesehen von Zähl- und Temp-Variablen für die Tauschoperation) kein zusätzlicher Speicherplatz benötigt!

(/2 Punkte)