

# ADS

## Zwischenprüfung

6. 12. 2016

Name, Vorname	Klasse	Punkte	Note
Frei, Jasmin	ITASTAZH	36.5	6

Zeit 45 Minuten

Punktzahl Total 40 Punkte

Hilfsmittel 2 Blatt A4-Papier, beidseitig (= 4 Seiten); keine elektronischen Hilfsmittel.

Abgabe Füllen Sie das Deckblatt aus.

Schreiben Sie **alle zusätzlichen Lösungsblätter mit Ihrem Namen, Vornamen und Ihrer Klasse** sowie der entsprechenden **Aufgabennummer** an.

**Geben Sie alle Aufgaben- und Lösungsblätter ab.**

Hinweise - Lesen Sie alle Aufgabenstellungen sorgfältig durch, bevor Sie mit der Bearbeitung der ersten Aufgabe beginnen.

- Die Aufgaben können in beliebiger Reihenfolge gelöst werden.

- Halten Sie sich nicht zulange an einer Aufgabe auf.

**Viel Glück!**

	Max. Punktzahl	Erreichte Punktzahl
Aufgabe 1: Hashing	10	
Aufgabe 2: Sortieren	10	
Aufgabe 4: Bäume	10	
Aufgabe 7: Allgemeine Fragen	10	
Total:	40	

**Aufgabe 1: Hashing [10 Punkte]**

16

Gegeben seien die Werte: 1238, 2341, 1200, 2318, 6789, 1236, 2346, 1231.

Teilaufgabe a)

Geben Sie die Belegung der Hashtabelle mit 10 Plätzen mit *linearem* Probing und der Hashfunktion  $h(x) = x \% 10$  an, wenn die Werte in der angegebenen Reihenfolge eingefügt werden.

0	1200
1	2341
2	6789
3	1231
4	
5	
6	1236
7	2346
8	1238
9	2318

Teilaufgabe b)

Gegeben seien (nochmals) die Werte: 1238, 2341, 1200, 2318, 6789, 1236, 2346, 1231.

Diesmal soll aber quadratisches Probing angewendet werden.

0	1200
1	2341
2	1231
3	6789
4	
5	
6	1236
7	2346
8	1238
9	2318

**Aufgabe 2: Sortieren [10 Punkte]** 10

Gegeben seien die untenstehenden Werte in einem Array

- a) Zeichnen Sie den Inhalt des Arrays nach dem ersten Durchgang des Bubblesort Algorithmus auf

13	4	12	2	19	4	2	53	3	5	1	23	41	2	4
----	---	----	---	----	---	---	----	---	---	---	----	----	---	---

4	12	2	13	4	2	19	3	5	1	23	41	2	4	53
---	----	---	----	---	---	----	---	---	---	----	----	---	---	----

- b) Selectionsort benötige zum Sortieren von 1000 Datensätzen 0.2 Millisekunden. Wie lange wird er für 1'000'000 Datensätzen benötigen?

$$k \cdot n \cdot \log_2(n) = 0,2 \text{ ms} \quad O(n \cdot \log_2(n))$$

$$k \cdot 1000 \cdot \log_2(1000) = 0,2 \text{ ms}$$

$$k = 0,000020069$$

$$k \cdot 1'000'000 \cdot \log_2(1'000'000) = 400 \text{ ms}$$

- c) Quicksort benötige zum Sortieren von 1000 Datensätzen ebenfalls 0.2 Millisekunden. Wie lange wird er für 1'000'000 Datensätzen benötigen?

$$k \cdot 1000^2 = 0,2 \text{ ms} \quad O(n^2)$$

$$k = 2 \cdot 10^{-10}$$

$$2 \cdot 10^{-10} \cdot 1'000'000^2 = 200 \text{ ms}$$

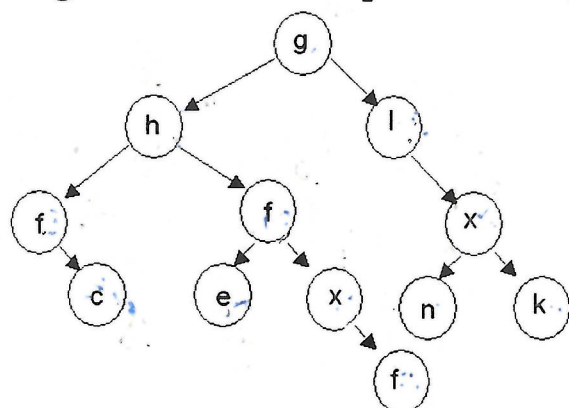
- e) InsertionSort benötige zum Sortieren von 1000 Datensätzen abermals 0.2 Millisekunden. Wie lange wird er für 1'000'000 Datensätzen benötigen, wenn die Daten sortiert sind?

$$k \cdot 1000^2 = 0,2 \text{ ms} \quad O(n^2)$$

$$k = 0,0000002$$

$$k \cdot 1'000'000 = 200 \text{ ms} \quad O(n)$$

## Aufgabe 3: Bäume [10 Punkte] 8



Gegeben seien die folgenden Ausgaben der besuchten Knoten:

1. cfefxfhnkxlg
2. ghfcfexflxnk
3. glhxffknxecf
4. ghlffxcexnkf
5. knxlfxfecfgh
6. fchxfxfglnxk
7. cffffeghklxxx
8. fcexnkffxhlg

a) Geben Sie an, bei welcher Ausgabe welche Traversierungsart angewendet wurde

Levelorder: 4

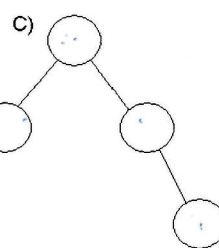
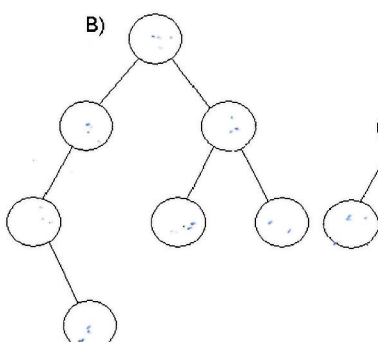
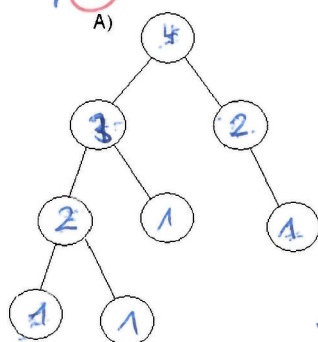
Inorder: 6

Preorder: 2

Postorder: 1

b) Welche der folgenden Bäume erfüllen das Kriterium der AVL-Ausgeglichenheit?

a, b, c



c) Welche Tiefe hat ein voller Binärbaum mit 12 Knoten?

4



## Aufgabe 4: Allgemeine Fragen [10 Punkte]

8 1/2

Bewertung: Je 0.5 Punkte für richtige Antwort, -0.5 Punkte für falsche Antwort, 0 Punkte für *weiss nicht*. Minimum: 0 Punkte.

Behauptung	wahr	falsch	weiss nicht
1. Ein Baum ist auch ein Graph.	X		
2. Jeder Knoten in einem binären Baum hat genau zwei Nachfolger.		X	
3. Die Postorder-Traversierung eines sortierten Baums gibt die Knoten in umgekehrt-alphabetischer Reihenfolge aus		X	
4. In einem binären Suchbaum hat das kleinste Element genau einen linken Nachfolger.		X	
5. Der Höhenunterschied der Teilbäume in einem AVL-Baum darf maximal 2 betragen		X	
6. Wenn man die equals-Methode überschreibt sollte man auch hashCode()-Methode überschreiben.	X		
7. Die Klasse String implementiert das Comparable Interface	X		
8. Ist der Belegungsgrad in einer Hashtabelle kleiner als 50%, ist das Einfügen von konstanter Ordnung.	X		
9. Für einen dichten Graphen ist die Adjazenzmatrix eine platzsparende Implementation	X	<del>X</del>	
10. Quicksort ist immer das effizienteste Sortierverfahren - auch im Best-Case.		X	
11. Eine sortierte Eingabe führt zu einem vollständig ausgeglichenem Baum..		X	
12. Die compareTo Methode eines Strings führt einen sprachen-/länderspezifischen String Vergleich durch.		X	
13. Divide-and-Conquer ("Teile und Herrsche") ist ein allgemeines Prinzip zur Entwicklung von effizienten Algorithmen	X		
14. Queues funktionieren nach dem LIFO-Prinzip.		X	
15. Vergisst man bei einem rekursiven Algorithmus die Abbruchbedingung, führt dies meist zu einem StackOverflow..	X		
16. Der schnellste Sortieralgorithmus vermeidet Rekursion.	X	<del>X</del>	
17. In einer doppelt verketteten Liste kommt jedes Element genau zweimal vor.		X	
18. Um den maximalen Fluss zu bestimmen, wird der Dijkstra Algorithmus angewendet.	X		
19. Es gibt zyklensfreien ungerichteten verbundene Graphen mit mehr als einem Knoten	<del>X</del>	X	
20. Die Grunddatenstruktur eines Ringbuffers ist ein Array	X		
21. Bonusfrage: der <i>Program Alert</i> bei der Mondlandung war 1202	X		

f

f