

Algorithmen und Datenstrukturen (ADS VO)	schriftliche Einzelprüfung	27.09.2013		1
--	----------------------------	------------	--	---

	$\begin{array}{r} 35 \\ + \square \end{array}$	$\begin{array}{r} 16 \\ + \square \end{array}$	$\begin{array}{r} 19 \\ + \square \end{array}$	$\begin{array}{r} 21 \\ + \square \end{array}$	$\begin{array}{r} 32 \\ + \square \end{array}$	$\begin{array}{r} 26 \\ + \square \end{array}$	$\begin{array}{r} 33 \\ + \square \end{array}$
18							
z_1	z_2	z_3	z_4	z_5	z_6	z_7	z_8

Aufgabe 1 [2]

Fügen Sie in obiger Tabelle in den leeren Kästchen, vor denen das Pluszeichen steht, die Ziffern Ihrer Matrikelnummer ein. Führen Sie die Additionen durch und ermitteln Sie die Zahlen **z_2** bis **z_8** . (**z_1** ist bereits mit dem fixen Wert 18 belegt.)

Aufgabe 2 [18]

- [9] Schreiben Sie **eine** Funktion in C++ mit einem Parameter n (vom Typ `int`), deren Laufzeitkomplexität **gleichzeitig** die Ordnungen $O(n^3)$, $\Omega(n)$ und $\Theta(n^2)$ hat.
- [9] Fügen Sie in nachfolgender Tabelle Kreuze an den Positionen ein, wo die in der Zeile angeführte Funktion von der in der Spalte angegebenen Ordnung ist.

$f(n)$	$O(n)$	$O(n^3)$	$O(\log n)$	$O(\log^2 n)$	$\Omega(n^2)$	$\Omega(\log n)$	$\Omega(\log^3 n)$	$\Theta(n^2)$	$\Theta(\log n)$	$\Theta(\log^3 n)$
n										
n^2										
n^3										
$\log n$										
$\log^2 n$										
$\log^3 n$										

Anmerkung: $\log^2 n = (\log n)(\log n)$

Aufgabe 3 [20]

Die Werte **z_1** bis **z_8** . (aus Aufgabe 1) seien in dieser Reihenfolge von links nach rechts in einem Array gespeichert. Sortieren Sie die Werte aufsteigend mit

- [8] Quicksort
- [4] Mergesort
- [8] Heapsort

Aufgabe 4 [20]

- [9] Fügen Sie die Werte **z_2** bis **z_8** aus Aufgabe 1 (in dieser Reihenfolge) in eine zu Beginn leere Hashtabelle der Länge 7 ein. Verwenden Sie als Hashfunktion **$h(k) = k \% 7$** und double hashing zur Kollisionsbehandlung. Die zweite Hashfunktion ist **$g(k) = k \% 5 + 1$** . Skizzieren Sie den Zustand der Hashtabelle nach jedem Einfügeschritt. (Anmerkung: Werte können mehrfach in der Tabelle gespeichert werden. Die Tabelle ist statisch, wird also nicht vergrößert!)
- [1] Löschen Sie den Wert **z_3** aus der Tabelle und skizzieren Sie den Zustand der Hashtabelle.
- [5] Geben Sie den Kollisionspfad (besuchte Indexpositionen) bei einer Suche nach dem Wert **z_8** an.
- [5] Geben Sie den Kollisionspfad (besuchte Indexpositionen) bei einer Suche nach dem Wert 50 an.

Algorithmen und Datenstrukturen (ADS VO)	schriftliche Einzelprüfung	27.09.2013		2
--	----------------------------	------------	--	---

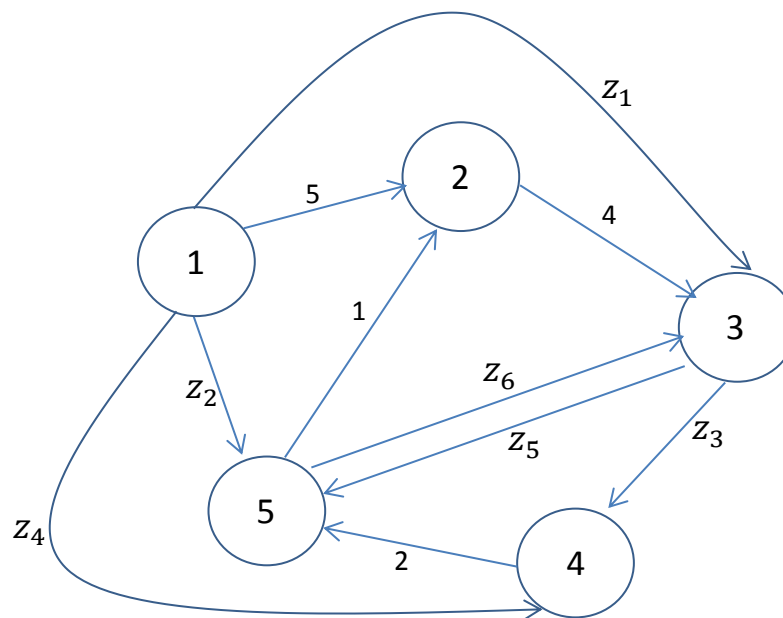
Aufgabe 5 [20]

- [8] Fügen Sie die Werte z_2 bis z_8 aus Aufgabe 1 (in dieser Reihenfolge) in einen zu Beginn leeren binären Suchbaum ein. Skizzieren Sie den Zustand des Baums nach jedem Einfügeschritt. (Anmerkung: Werte können mehrfach im Baum gespeichert werden.)
- [4] Geben Sie in C++ ähnlicher Notation die Definition einer Datenstruktur für einen binären Suchbaum an.
- [4] Geben Sie in C++ ähnlicher Notation eine Definition einer Funktion an, die den binären Suchbaum depth first traversiert und alle gespeicherten Werte ausgibt.
- [4] Bestimmen Sie die Laufzeitkomplexität Ihrer Traversierungsfunktion abhängig von der Anzahl n der im Suchbaum gespeicherten Werte in Θ -Notation.

Aufgabe 6 [20]

Gegeben ist der folgende gerichtete Graph

(die Werte z_1 bis z_6 sind aus Aufgabe 1 zu übernehmen):



- [3] Geben Sie die Adjazenzmatrix des Graphen an.
- [3] Skizzieren Sie die Adjazenzliste des Graphen.
- [10] Bestimmen Sie mit dem Algorithmus von Dijkstra die jeweils kürzesten Wege vom Knoten 1 zu allen anderen Knoten des Graphen.
- [4] Ist für den Dijkstra-Algorithmus eher die Verwendung einer Adjazenzmatrix oder einer Adjazenzliste vorteilhaft? Begründen Sie Ihre Aussage.