

Algorithmen und Datenstrukturen (ADS VO)	schriftliche Einzelprüfung	18.01.2019		1
--	----------------------------	------------	--	---

13	14	32	15	18	23	39	34
+ <input type="text"/>	+ <input type="text"/>	+ <input type="text"/>	+ <input type="text"/>	+ <input type="text"/>	+ <input type="text"/>	+ <input type="text"/>	+ <input type="text"/>
$z_1$	$z_2$	$z_3$	$z_4$	$z_5$	$z_6$	$z_7$	$z_8$

### 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  $z_1$  bis  $z_8$ .

### Aufgabe 2 [18]

- [10] Erstellen Sie in C++-ähnlichem Pseudocode eine rekursive Funktion  $f$  mit einem ganzzahligen Parameter  $n$ , auf die das Mastertheorem anwendbar ist und deren Laufzeit in  $\Theta(n^2 \log n)$  liegt.
- [3] Zeigen Sie mit Hilfe des Mastertheorems, dass  $f$  die gewünschte Laufzeit hat.
- [5] Erstellen Sie in C++-ähnlichem Pseudocode eine Funktion  $g$  mit einem ganzzahligen Parameter  $n$ , die  $f$  aufruft und deren Laufzeit in  $\Theta(n^2 (\log n)^3)$  liegt.

### 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] CountingSort
- [4] Mergesort
- [8] Heapsort

Geben Sie jeweils alle benötigten Zwischenschritte so genau an, dass der Ablauf des Algorithmus klar ersichtlich wird.

### 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.
- [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.

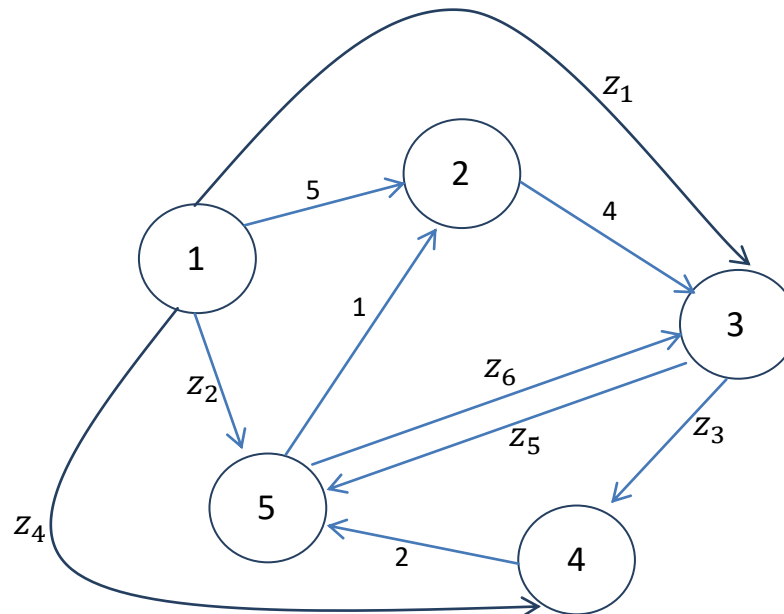
### Aufgabe 5 [20]

- [4] Fügen Sie die Werte  $z_1$  bis  $z_8$  aus Aufgabe 1 (in dieser Reihenfolge) in einen zu Beginn leeren binären Suchbaum ein (Werte können im Baum eventuell mehrfach gespeichert sein). Skizzieren Sie den Zustand des Baums nach jedem Einfügeschritt.
- [4] Geben Sie in C++ ähnlicher Notation die Definition einer Datenstruktur für einen binären Suchbaum an.
- [8] Geben Sie in C++ ähnlicher Notation eine Definition einer Funktion oder Methode an, die die Höhe eines binären Suchbaums ermittelt.
- [4] Bestimmen Sie die Laufzeitkomplexität Ihrer Funktion abhängig von der Anzahl  $n$  der im Suchbaum gespeicherten Werte in O-Notation. Begründen Sie Ihr Ergebnis kurz.

### 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.