

Algorithmen und Datenstrukturen ADS VO	schriftliche Einzelprüfung	09.12.2016	«MATRIKELNR» «NACHNAME» «VORNAME»	1
--	-------------------------------	------------	-----------------------------------	---

	<b>21</b>	<b>38</b>	<b>13</b>	<b>15</b>	<b>40</b>	<b>27</b>	<b>48</b>
	+ <input type="text"/>	+ <input type="text"/>	+ <input type="text"/>	+ <input type="text"/>	+ <input type="text"/>	+ <input type="text"/>	+ <input type="text"/>
<b>45</b>							
<b>z<sub>1</sub></b>	<b>z<sub>2</sub></b>	<b>z<sub>3</sub></b>	<b>z<sub>4</sub></b>	<b>z<sub>5</sub></b>	<b>z<sub>6</sub></b>	<b>z<sub>7</sub></b>	<b>z<sub>8</sub></b>

### 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<sub>2</sub>** bis **z<sub>8</sub>**. (**z<sub>1</sub>** ist bereits mit dem fixen Wert 45 belegt.)

### Aufgabe 2 [18]

Berechnen Sie die Konstanten **a** = **z<sub>6</sub>**%10 + 1, **b** = **z<sub>7</sub>**%2 + 2, **c** = **z<sub>8</sub>**%3. Geben Sie ein Programm in C++ ähnlicher Notation an, bei dessen Laufzeituntersuchung mit dem Mastertheorem die Formelparameter a,b und c die ermittelten Werte annehmen. Bestimmen Sie die Laufzeitkomplexität Ihres Programms.

Anmerkung: Bei Bedarf kann Ihr Programm auch aus mehreren Funktionen bestehen, die einander aufrufen.

### Aufgabe 3 [20]

Die Werte **z<sub>1</sub>** bis **z<sub>8</sub>**. (aus Aufgabe 1) seien in dieser Reihenfolge von links nach rechts in einem Array gespeichert. Sortieren Sie die Werte aufsteigend mit

- [10] Quicksort (Verwenden Sie als Pivotelement immer den letzten - ganz rechten - Wert).
- [6] Heapsort
- [4] Insertion Sort

Geben Sie alle notwendigen Schritte so genau an, dass die Arbeitsweise des Algorithmus klar ersichtlich wird.

### Aufgabe 4 [20]

- [5] Fügen Sie die Werte **z<sub>1</sub>** bis **z<sub>8</sub>** aus Aufgabe 1 (in dieser Reihenfolge) in einen zu Beginn leeren binären Suchbaum ein. Skizzieren Sie den Zustand des Baums nach jedem Einfügeschritt.  
Anmerkung: der Baum kann Werte mehrfach enthalten.
- [5] Geben Sie in C++ ähnlicher Notation die Definition einer Datenstruktur für einen binären Suchbaum an.
- [5] Geben Sie in C++ ähnlicher Notation eine Definition einer Funktion an, die das Maximum und das Minimum der im Baum gespeicherten Werte ermittelt und ausgibt.

[5] Geben Sie an, welche Knoten in welcher Reihenfolge Ihre Funktion traversiert, wenn sie für den unter Teilaufgabe a erstellten Baum aufgerufen wird. Handelt es sich dabei um eine preorder Traversierung, eine postorder Traversierung, eine inorder Traversierung oder um eine andere Art der Traversierung?

Algorithmen und Datenstrukturen ADS VO	schriftliche Einzelprüfung	09.12.2016	«MATRIKELNR» «NACHNAME» «VORNAME»	2
--	-------------------------------	------------	-----------------------------------	---

### Aufgabe 5 [20]

- [10] Fügen Sie die Zahlen  $z_1$  bis  $z_7$  (aus Beispiel 1) in dieser Reihenfolge in eine ursprünglich leere Hashtabelle der Größe 8 ein. Verwenden Sie die Hashfunktion  $h(x) = x \% 8$  und zur Kollisionsbehandlung double Hashing mit  $g(x) = 9$  als zweiter Hashfunktion. Geben Sie den Zustand der Hashtabelle nach jeder Einfügeoperation an.
- [4] Geben Sie den Kollisionspfad an, der durchsucht wird, wenn versucht wird, in der nach a) befüllten Hashtabelle zusätzlich  $z_8$  (aus Beispiel 1) einzufügen.
- [2] Wozu wird beim double Hashing die Markierung „wiederfrei“ verwendet?
- [2] Warum ist es in diesem Beispiel kein Problem, dass die Größe der Tabelle (8) keine Primzahl ist?
- [2] Warum ist  $g(x)$  keine ideale Funktion für die Kollisionsbehandlung?

### Aufgabe 6 [20]

Gegeben ist die folgende Adjazenzmatrix, die die Kosten der Verbindungen zwischen den Knoten eines gerichteten Graphen beschreibt:

$$\begin{pmatrix} 0 & 0 & z_8 & 4 & z_6 \\ z_5 & 0 & 0 & z_3 & 2 \\ z_2 & z_4 & 0 & z_1 & z_7 \\ 0 & 0 & 3 & 0 & z_5 \\ 0 & 1 & 0 & z_2 & 0 \end{pmatrix}$$

- [2] Skizzieren Sie den Graphen, der durch diese Adjazenzmatrix beschrieben wird.
- [10] Bestimmen Sie mit dem Algorithmus von Dijkstra die kürzesten Wege vom Knoten 3 zu allen anderen Knoten des Graphen (Dabei entspricht Knoten 3 dem Knoten der dritten Zeile/Spalte in der Adjazenzmatrix).
- [8] Bestimmen Sie mit dem Algorithmus von Kruskal einen minimal spannenden Baum des Schattens des Graphen. (Sie erhalten den Schatten des Graphen, indem Sie die Richtungen der Kanten vernachlässigen. Werden dann zwei Knoten durch zwei oder mehr Kanten verbunden, so werden diese Kanten zu einer zusammengefasst. Anders ausgedrückt: Zwei Knoten x und y im Schatten sind genau dann durch eine ungerichtete Kante verbunden, wenn im ursprünglich gerichteten Graphen zumindest eine der Kanten von x nach y oder von y nach x existiert. Als Gewicht der ungerichteten Kante wählen sie jeweils das Minimum aller durch sie repräsentierten gerichteten Kanten.) Notieren Sie alle Zwischenschritte so genau, dass klar ist, wann welche Kante zum spannenden Baum hinzugefügt wird.