

Algorithmen und Datenstrukturen 1 (ADS VU)	schriftliche Einzelprüfung (Haupttermin)	30.06.2020		1
--------------------------------------------------	------------------------------------------------	------------	--	---

22	25	29	32	19	28	19	24
+	+	+	+	+	+	+	+
$z_1$	$z_2$	$z_3$	$z_4$	$z_5$	$z_6$	$z_7$	$z_8$

### 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]

Gegeben seien folgende Funktionen:

```
void well_known(int n, int digit) {
    for (int i=n; i>0; i=i/2)
        for (int j=0; j<n; j++)
            cout << "easy";
}
```

```
void known(int n, int digit) {
    if (digit==0) return
    for (int i=0; i<n; ++i)
        known(n,digit-1);
}
```

```
void unknown(int n, int digit) {
    for (int i=0; i<3; ++i) known(n, digit);
    for (int i=0; i<9; ++i) unknown(n/digit, digit);
}
```

Setzen Sie für den Parameter **digit** den Wert  $z_6 \% 3 + 1$  ein und finden Sie die Laufzeitabschätzungen in Theta-Notation (abhängig von n) für:

- [4] die Funktion **well\_known**
- [6] die Funktion **known**
- [8] und die Funktion **unknown**.

### 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
- [8] Heapsort
- [4] Counting Sort (**Achtung:** Bei dieser Aufgabe bitte jeweils Wert modulo 10 verwenden, z. B.  $z_1 \% 10$ ,  $z_2 \% 10$  usw.)

### 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  $g(k) = k \% 3 + 2$ . Skizzieren Sie den Zustand der Hashtabelle nach jedem Einfügeschritt. (Werte können im Kontext dieser Aufgabe mehrfach in der Hashtabelle gespeichert werden.)
- [1] Löschen Sie den Wert  $z_5$  aus der Tabelle, die im Punkt a) erstellt wurde und skizzieren Sie den Zustand der Hashtabelle.
- [5] Geben Sie den Kollisionspfad (besuchte Indexpositionen) bei einer Suche nach dem Wert  $z_7$  in der nach b) entstandenen Hashtabelle an.
- [5] Geben Sie den Kollisionspfad (besuchte Indexpositionen) bei einer Suche nach dem Wert 42 in der nach b) entstandenen Hashtabelle an.

Algorithmen und Datenstrukturen 1 (ADS VU)	schriftliche Einzelprüfung (Haupttermin)	30.06.2020		2
--------------------------------------------------	------------------------------------------------	------------	--	---

### Aufgabe 5 [20]

- [8] 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. (Methodendeklarationen oder sogar –definitionen sind **nicht** notwendig. Es reichen die Instanzvariablen.)
- [4] Geben Sie in C++ ähnlicher Notation eine Definition einer Funktion an, die den in b) definierten binären Suchbaum depth first traversiert und alle gespeicherten Werte ausgibt.
- [4] Notieren Sie die Ausgabe Ihrer Funktion aus c), 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?

### Aufgabe 6 [20]

Gegeben ist die folgende Adjazenzmatrix mit Wegekosten für einen gerichteten Graphen (die Werte  $z_1$  bis  $z_8$  sind aus Aufgabe 1 zu übernehmen):

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

- [2] Skizzieren Sie den gerichteten Graphen.
- [10] Bestimmen Sie mit dem Algorithmus von Dijkstra die jeweils kürzesten Wege vom Knoten 1 (erste Zeile, erste Spalte der Matrix) zu allen anderen Knoten des Graphen.
- [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 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.)