

Algorithmen und Datenstrukturen (ADS VO)	schriftliche Einzelprüfung	07.11.2014		1
--	----------------------------	------------	--	---

	16	15	28	20	32	24	14
	+ <input type="text"/>	+ <input type="text"/>	+ <input type="text"/>	+ <input type="text"/>	+ <input type="text"/>	+ <input type="text"/>	+ <input type="text"/>
12							
z₁	z₂	z₃	z₄	z₅	z₆	z₇	z₈

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 **z₂** bis **z₈**. (**z₁** ist bereits mit dem fixen Wert 12 belegt.)

Aufgabe 2 [18]

```
void f(int n) {
    for (int i=n; i>0; i=i/2)
        for (int j=1; j<n; j=j*4)
            for (int k=n; n>0; n=n-3);
}

void g(int n) {
    if (n==0) return;
    for (int i=3; i<2*n; ++i);
    g(n-2);
}

void h(int n, int digit) {
    if (n==0) return;
    for (int i=0; i<n; ++i) g(n);
    for (int i=0; i<digit; ++i) h(n/2, digit);
}
```

Nehmen Sie an, dass für den Parameter digit der Wert $|z_8 - z_7| + 2$ übergeben wird und finden Sie Laufzeitabschätzungen in Theta-Notation (abhängig von n) für

- [4] die Funktion f,
- [6] die Funktion g,
- [8] und die Funktion h.

Anmerkung: $|z_8 - z_7|$ ist der nicht negative Betrag der Differenz der beiden Werte.

Aufgabe 3 [20]

Die Werte **z₁** bis **z₈**. (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] Counting Sort
- [4] Selection Sort

Aufgabe 4 [20]

- [10] Fügen Sie die Zahlen **z₁** bis **z₈** (aus Beispiel 1) in aufsteigender Reihenfolge der Indizes in einen zu Beginn leeren Max-Heap ein. Geben Sie den Zustand des Heaps nach jeder Einfügeoperation sowohl in der Baumdarstellung, als auch in der Darstellung als Feld an.
- [6] Löschen Sie danach dreimal das Wurzelement. Geben Sie den Zustand des Heaps nach jeder Löschoption sowohl in der Baumdarstellung, als auch in der Darstellung als Feld an.
- [2] Welche Laufzeitordnungen haben die Operationen Einfügen, Löschen und Maximum-Suchen in einem Max-Heap?
- [2] Welche Laufzeitordnung hat Counting Sort?

Algorithmen und Datenstrukturen (ADS VO)	schriftliche Einzelprüfung	07.11.2014		2
--	----------------------------	------------	--	---

Aufgabe 5 [20]

- a) [10] Fügen Sie die Zahlen z_1 bis z_8 (aus Beispiel 1) in aufsteigender Reihenfolge der Indizes in eine ursprünglich leere Hashtabelle der Größe 7 ein. Verwenden Sie die Hashfunktion $h(x) = x \% 7$ und zur Kollisionsbehandlung double Hashing mit $g(x) = x \% 3 + 3$ als zweiter Hashfunktion. Geben Sie den Zustand der Hashtabelle nach jeder Einfügeoperation an.
- b) [10] Fügen Sie die Zahlen 2, 3, 4, 7 und 5 in dieser Reihenfolge in eine Hashtabelle mit dem Verfahren Linear Hashing ein. Starten Sie mit einer leeren Tabelle mit Blockgröße 2 ($b = 2$) und nur einem Datenblock ($d = 0$). Skizzieren Sie den Zustand der Tabelle nach jedem Einfügeschritt.
- ($2 \equiv 00010$, $3 \equiv 00011$, $4 \equiv 00100$, $5 \equiv 00101$, $7 \equiv 00111$)

Aufgabe 6 [20]

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

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 42 \\ z_4 & 0 & 0 & 0 & 0 \\ 0 & z_5 & 5 & z_6 & 2 \\ z_3 & z_8 & 0 & 0 & 0 \\ z_2 & 0 & 1 & z_7 & 9 \end{pmatrix}$$

- a) [2] Skizzieren Sie den Graphen, der durch diese Adjazenzmatrix beschrieben wird.
- b) [10] Bestimmen Sie mit dem Algorithmus von Dijkstra die kürzesten Wege vom Knoten 1 zu allen anderen Knoten des Graphen (Dabei entspricht Knoten 1 dem Knoten der ersten Zeile/Spalte in der Adjazenzmatrix).
- c) [8] Entfernen Sie eine möglichst kleine Anzahl von Kanten, sodass der Graph topologisch sortierbar wird (führen Sie genau an, welche Kanten entfernt werden müssen) und führen Sie eine topologische Sortierung durch (es reicht, die Abfolge der Knoten anzugeben, Sie müssen nicht den Graphen für das Resultat zeichnen).