

Aufgabe 1 [20]

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);
}
```

Nehmen Sie an, dass für den Parameter *digit* die zweite Stelle Ihrer Matrikelnummer (von rechts gezählt) plus 1 übergeben wird und finden Sie Laufzeitabschätzungen in Theta-Notation (abhängig von n) für

- [4] die Funktion *well_known*,
- [6] die Funktion *known*,
- [10] und die Funktion *unknown*.

Aufgabe 2 [20]

- [16] Die Zahlen 13, 5, 14, 10, 1, 9, 12, 4, 7, 15, 2, 6, 11, 3 und 8 seien in dieser Reihenfolge in einem Array gespeichert. Sortieren Sie die Zahlen aufsteigend mit dem Quicksort-Verfahren. Wählen Sie als Pivotelement immer die letzte (ganz rechte Zahl). Geben Sie alle benötigten Zwischenschritte so genau an, dass der Ablauf des Algorithmus' klar ersichtlich wird.
- [4] Nehmen Sie an, es wären die Ziffern Ihrer Matrikelnummer mit Quicksort zu sortieren. Als Pivotelement würde immer die letzte (ganz rechte) Ziffer gewählt. Wie müssen die Ziffern ursprünglich angeordnet sein, damit Quicksort die schlechteste Laufzeit erreicht?

Aufgabe 3 [20]

Addieren Sie zu Ihrer Matrikelnummer die Zahl 38769451. Nehmen Sie an, die Ziffern der so erhaltenen Zahl wären (in der gleichen Reihenfolge in der sie in der Zahl auftreten) in einem Array gespeichert. Sortieren Sie dieses Array

- [8] mit Heapsort
- [4] mit Mergesort
- [8] mit Counting Sort

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

Aufgabe 4 [20]

- [8] Fügen Sie die Zahlen 1, 26, 15, 11 und 42 in dieser Reihenfolge in eine ursprünglich leere Hashtabelle der Größe 7 ein. Verwenden Sie die Hashfunktion $h(x) = x\%7$ und zur Kollisionsbehandlung double Hashing mit $g(x) = x\%10 + 1$ (also die letzte Ziffer plus 1) als zweiter Hashfunktion. Geben Sie den Zustand der Hashtabelle nach jeder Einfügeoperation an.
- [2] Löschen Sie die Zahl 15 aus der Hashtabelle (geben Sie an, welche Positionen der Hashtabelle Sie besuchen müssen, um die zu löschende Zahl zu finden, sowie den Zustand der Hashtabelle nach dem Löschen).
- [4] Suchen Sie (nach dem Löschen) die Zahlen 42 und 35 in der Hashtabelle (geben Sie für jede Zahl an, welche Positionen der Hashtabelle Sie besuchen müssen, und ob die Suche erfolgreich ist, oder nicht).
- [2] Wozu wird beim double hashing die Markierung "wiederfrei" verwendet?
- [2] Was ist der grundlegende Unterschied zwischen statischen und dynamischen Hashverfahren?
- [2] Welche dynamischen Hashverfahren haben Sie in der Vorlesung kennen gelernt (Aufzählung der Namen reicht aus)?

Algorithmen und Datenstrukturen (PI.ADS.AD.VO)	schriftliche Einzelpruefung	01.12.2008		2
--	--------------------------------	------------	--	---

Aufgabe 5 [20]

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

$$\begin{pmatrix} 0 & z3 & z2 & z1 & 0 \\ 0 & 0 & z4 & 0 & z7 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & z5 & 0 & z6 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

- [2] Ersetzen Sie in der Adjazenzmatrix die Gewichte $z1$ bis $z7$ durch Werte, die Sie aus Ihrer Matrikelnummer wie folgt ermitteln: z_i ergibt sich aus der i -ten Stelle der Matrikelnummer (von rechts beginnend nummeriert) plus 1. Für die Matrikelnummer 1234567 wäre $z2$ beispielsweise 7 ($=6+1$).
Skizzieren Sie den Graphen, der durch diese Adjazenzmatrix beschrieben wird.
- [10] Bestimmen Sie mit dem Algorithmus von Dijkstra den kürzesten Weg vom Knoten 1 zum Knoten 5 des Graphen (Dabei entspricht Knoten 1 dem Knoten der ersten Zeile/Spalte in der Adjazenzmatrix und Knoten 5 der fünften Zeile/Spalte).
- [8] Entfernen Sie eine möglichst kleine Anzahl von Kanten, sodass der Graph topologisch sortierbar wird, und führen Sie eine topologische Sortierung durch (es reicht, die Abfolge der Knoten anzugeben, Sie müssen nicht den Graphen zeichnen).