

Aufgabe 1 [20]

Gegeben seien folgende Funktionen

```
void done(int n) {
    for (int i=n; i>0; i=i/2)
        for (int j=1; j<n; j=j*4)
            cout<<"easy";
}
```

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

```
void doing(int n, int digit) {
    doing(n/digit, digit);
    for (int i=0; i<n; ++i) todo(n);
    doing(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 *done*,
- [6] die Funktion *todo*,
- [10] und die Funktion *doing*.

Aufgabe 2 [20]

Addieren Sie zu Ihrer Matrikelnummer die Zahl 14215963. Die Ziffern der Summe seien in der Reihenfolge von links nach rechts in einem Array gespeichert. Sortieren Sie die Ziffern aufsteigend mit

- [10] Quicksort. Wählen Sie als Pivotelement immer die letzte (ganz rechte Ziffer).
- [6] Counting Sort
- [4] Mergesort

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

Aufgabe 3 [20]

- [10] Fügen Sie die Zahlen 1, 26, 15, 11 und 42, sowie die letzten drei Ziffern Ihrer Matrikelnummer (von rechts nach links) in dieser Reihenfolge in einen zu Beginn leeren Min-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 Minimum-Suchen in einem Min-Heap?
- [2] Welche Laufzeitordnung hat Heapsort?

Aufgabe 4 [20]

- [10] Fügen Sie die Zahlen 7, 0, 4, 21 und 14 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 $g(x) = x\%10 + i$ (also die letzte Ziffer plus i, dabei wählen Sie für i den Wert der letzten Ziffer Ihrer Matrikelnummer plus 1) als zweiter Hashfunktion. Geben Sie den Zustand der Hashtabelle nach jeder Einfügeoperation an.
- [2] Löschen Sie die Zahl 0 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).
- [6] Suchen Sie (nach dem Löschen) die Zahlen 7,14 und 0 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] Was ist der grundlegende Unterschied zwischen statischen und dynamischen Hashverfahren?

Aufgabe 5 [20]

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

$$\begin{pmatrix} 0 & z3 & 0 & z2 & z1 \\ 0 & z4 & 0 & z7 & 0 \\ 1 & 0 & 5 & 0 & 1 \\ 0 & 0 & z5 & 0 & z6 \\ 1 & 0 & 2 & 0 & 9 \end{pmatrix}$$

- [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 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).
- [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).