

Aufgabe 1 [20]

Gegeben seien folgende Funktionen

```
void veni(int n) {
    for (int i=3; i<n; ++i)
        for (int j=0; j<i; ++j)
            veni(n-1);
}

void vidi(int n) {
    for (int i=1; i<n; i=i*5)
        for (int j=n; j>0; j/=2);
}

void vici(int n, int digit) {
    if (n==0) return;

    for (int i=0; i<digit; ++i)
        vici(n/2, digit);
}
```

Nehmen Sie an, dass für den Parameter *digit* die zweite Stelle Ihrer Matrikelnummer (von rechts gezählt) plus 10 übergeben wird und finden Sie Laufzeitabschätzungen in Theta-Notation (abhängig von *n*) für

- [5] die Funktion *veni*,
- [5] die Funktion *vidi*,
- [10] und die Funktion *vici*.

Aufgabe 2 [20]

Addieren Sie zu Ihrer Matrikelnummer die Zahl 364258817. Die Ziffern der Summe seien in der Reihenfolge von links nach rechts in einem Array gespeichert. Sortieren Sie die Ziffern aufsteigend mit

- [9] Quicksort. Wählen Sie als Pivotelement immer die erste (ganz linke Ziffer).
- [9] Heapsort

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

- [2] Geben Sie die durchschnittliche und die schlechteste Laufzeitordnung von Quicksort an.

Aufgabe 3 [15]

- [4] Erstellen Sie einen Expression Tree für den Ausdruck $7+5*2-(3+8)*(4-1)+9/6$.
- [9] Geben Sie die drei Reihenfolgen an, in denen die Knoten bei inorder-, preorder- und postorder-Traversierung bearbeitet werden.
- [2] In welcher Reihenfolge werden die Knoten bearbeitet, wenn der Baum mit bfs traversiert wird?

Aufgabe 4 [25]

- [20] Beschreiben Sie die grundlegenden Ideen, die allen Hashverfahren zugrunde liegen. Beschreiben Sie, wie Kollisionen mit dem double-hashing Verfahren behandelt werden. Wozu wird die Markierung wiederfrei benötigt? Demonstrieren Sie die Notwendigkeit von wiederfrei-Markierungen anhand eines von Ihnen gewählten konkreten Beispiels. Welche Ordnungen haben die Operationen Einfügen, Suchen und Löschen in einer Hashtabelle mit double-hashing?
- [5] Beschreiben Sie den Unterschied zwischen statischen und dynamischen Hashverfahren. Welche dynamischen Hashverfahren kennen Sie (Eine Aufzählung der Namen reicht aus)?

Aufgabe 5 [20]

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

$$\begin{pmatrix} 0 & 20 & 21 & 22 & 0 \\ 0 & 0 & z1 & 0 & 25 \\ 0 & z2 & 0 & z3 & 50 \\ z4 & 0 & z5 & 0 & z6 \\ 0 & 0 & z7 & 0 & 0 \end{pmatrix}$$

- a. [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.
- 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] Bestimmen Sie mit dem Algorithmus von Prim 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.) Notieren Sie alle Zwischenschritte so genau, dass klar ist, wann welche Kante zum spannenden Baum hinzugefügt wird.