Aufgabe 1 [20]

Gegeben sind folgende Funktionen (a, b und c seien vordefinierte Konstanten):

```
void f(int n) {
  if (!n) return;
  for (int i=0; i<a; i++)
    f(n/b);
  g(n,c);
}

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

Berechnen Sie die Laufzeit der Funktion f in Θ-Notation abhängig von n. Belegen Sie dabei die Konstanten a, b und c jeweils mit den um 2 erhöhten Werten der letzten drei Stellen Ihrer Matrikelnummer (also Matrikelnummer 1118059 ergibt z.B.: a=2 b=7 und c=11).

(Hinweis: Erstellen Sie Rekurrenzgleichungen für die Laufzeiten von g bzw. f und lösen Sie diese mittels fortgesetztem Einsetzen bzw. Master Theorem.)

Aufgabe 2 [20]

Addieren Sie zu Ihrer Matrikelnummer die Zahl 679845132. Die Ziffern der Summe seien in der Reihenfolge von links nach rechts in einem Array gespeichert. Sortieren Sie die Ziffern aufsteigend mit

a. [8] Counting Sort.

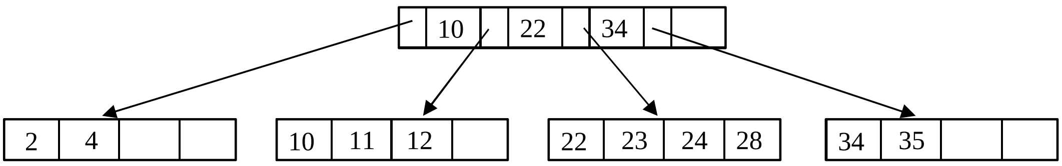
b. [8] Quicksort.

c. [4] Selection Sort.

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

Aufgabe 3 [20]

Gegeben ist der folgende $B^+ - Baum$ der Ordnung 2.



Fügen Sie die Werte 9, 14, 29 und 15 (in dieser Reihenfolge ein und löschen Sie danach die Werte 10 und 15 (in dieser Reihenfolge).

Skizzieren Sie den Zustand des Baumes nach jeder Einfüge- bzw. Löschoperation.

Aufgabe 4 [20]

Gegeben ist der arithmetische Ausdruck 6*2-(7+1)+9/(4-5*(8+3))

a. [4] Geben Sie einen Expression Tree für diesen Ausdruck an.

b. [6] Geben Sie die verschiedenen Reihenfolgen an, in denen die Knoten besucht werden, wenn Sie den Baum inorder, preorder und postorder traversieren.

c. [5] Nehmen Sie an, es wäre ein Expression Tree für n binäre Operationen zu erstellen. Welche maximale bzw. minimale Höhe des Expression Trees ist zu erwarten?

d. [5] Geben Sie in C++-artigem Pseudocode eine mögliche Klassendefinition für die Datenstruktur eines binären Baumes an. (Es müssen keine Methoden angeführt werden!)

Aufgabe 5 [20]

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

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

a. [2] Ersetzen Sie in der Adjazenzmatrix die Gewichte z1 bis z7 durch Werte, die Sie aus Ihrer Matrikelnummer wie folgt ermitteln: zi 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 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).

c. [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).