Algorithmen und Datenstrukturen 1	schriftliche Einzelpruefung	25.03.2004		1

Aufgabe 1 [2.0]

a. Gegeben ist folgendes rekursive Programm:

```
int f(n) {
    int s = 0;
    if(n<=1) return 1;
    else s = s + f(n/4) + f(n/4);
    return s;
}
```

[0.2] Entwickeln Sie die *Rekurrenzgleichung*, die das Laufzeitverhalten der Funktion f(n) beschreibt.

[0.5] Bestimmen Sie das *asymptotische Laufzeitverhalten* der Funktion f(n) mit Hilfe des Master Theorems.

b. [0.4] Definieren Sie den Begriff O(n) bei der Laufzeitabschätzung eines Programms?

c. [0.4] Erläutern Sie, warum die algorithmische Lücke beim Sortieren basierend auf Elementvergleichen geschlossen ist.

d. [0.5] Skizzieren Sie den Programmgraphen des obigen Programms und berechnen Sie dessen zyklomatische Komplexität nach der Metrik von McCabe.

Aufgabe 2 [2.0]

a. [1.1] Fügen Sie die Werte 5, 8, 3, 4, 6, 7, 1, 2, 9, 11, 10 in dieser Reihenfolge in einen ursprünglich leeren *2-3-4 - Baum* ein.

b. [0.9] Löschen Sie die Werte 5, 6 und 4 (in dieser Reihenfolge) in dem oben erstellten 2-3-4 Baum.

Geben Sie alle Zwischenschritte bei der Veränderung des Baums an.

Aufgabe 3 [1.6]

Die Werte 5, 7, 8, 1, 6, 3, 2, 4 sind in dieser Reihenfolge in einem Vektor der Länge 8 gespeichert.

Sortieren Sie diesen Vektor jeweils mit dem Verfahren

a. [0.5] Countingsort und

b. [0.5] Bucketsort.

Geben Sie alle Zwischenschritte bei den Sortievorgängen an.

c. [0.6] Erläutern Sie informell, warum die obigen Sortierverfahren von der Ordnung O(n) sind.

Aufgabe 4 [1.5]

a. [0.5] Tragen Sie die folgenden Zeichenketten (Strings) in die Datenstruktur eines Tries ein. Skizzieren Sie den schrittweisen Aufbau des digitalen Suchbaumes (Trie).

- SAMBA
- SALSA
- CCC
- TANGO
- PASO
- PASODOBLE

Stellen Sie die Struktur des Tries graphisch dar und geben Sie alle Zwischenschritte an.

b. [0.2] Geben Sie eine geeignete Datenstruktur in C++ zur Speicherung eines Tries an.

c. [0.8] Skizzieren Sie in Pseudocode einen Algorithmus zur Suche aller gespeicherten Strings im Trie, die mit einem gegebenen Teilstring (Praefix) beginnen.

Dokumentieren Sie Ihr Programm ausführlich.

Aufgabe 5 [2.0]

a. [1.0] Fügen Sie die Werte 8, 7, 6, 5, 4, 3, 2 und 1 in dieser Reihenfolge in einen ursprünglich leeren Heap ein (das Minimum ist in der Wurzel gespeichert). Stellen Sie den Heap als binären Baum dar.

b. [0.6] Löschen Sie 3x das Minimum.

Geben Sie alle Zwischenschritte bei der Veränderung des Baums an.

c. [0.4] Geben Sie die Aufwände in O-Notation für die Heap-Operationen Einfügen, Löschen und Zugriff auf das Minimum an und leiten Sie diese informell ab.

Aufgabe 6 [0.9]

Erläutern Sie die Algorithmenentwurfsverfahren *greedy, divide-and-conquer* und *dynamic programming* jeweils anhand ihrer

- zugrundeliegenden Ideen, und
- geben Sie ein Beispiel.