Prof. Dr. Ir. Joost-Pieter Katoen

Christian Dehnert, Jonathan Heinen, Thomas Ströder, Sabrina von Styp

Hinweise:

- Die Übungsblätter sollen in Gruppen von je 3 Studierenden aus der gleichen Kleingruppenübung bearbeitet werden.
- Die Lösungen müssen bis Montag, den 23. April um 11:00 Uhr in den entsprechenden Übungskasten eingeworfen werden. Sie finden die Kästen am Eingang Halifaxstr. des Informatikzentrums (Ahornstr. 55).
- Namen und Matrikelnummern der Studenten sowie die Nummer der Übungsgruppe sind auf jedes Blatt der Abgabe zu schreiben. Heften bzw. tackern Sie die Blätter!

Aufgabe 1 (Baumeigenschaften): (2+2+2 Punkte)

Beweisen Sie folgende in der Vorlesung eingeführte Fakten für Binärbäume:

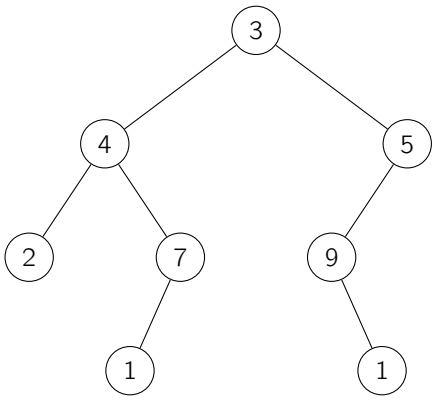
a) Ein Binärbaum enthält höchstens 2^d Knoten in Ebene d.

b) Ein Binärbaum mit Höhe h kann maximal $2^{h+1} - 1$ Knoten enthalten.

c) Ein Binärbaum mit n Knoten hat mindestens die Höhe $\lceil \log_2(n+1) \rceil - 1$.

Aufgabe 2 (Baumtraversierung): (1+2+2 Punkte)

a) Geben Sie jeweils das Ergebnis der in-, pre- und post-order Traversierung des folgenden Baumes an:



b) Bestimmen Sie zu den folgenden Paaren von Linearisierungen den jeweils zugehörigen Baum:

(i) in-order: 5 8 4 7 3 1 6 9 2
pre-order: 3 4 5 8 7 2 6 1 9

(ii) in-order: 5 1 2 4 6 7 3 9 8
post-order: 5 2 4 1 3 8 9 7 6

c) Geben Sie ein minimales Beispiel für zwei unterschiedliche Bäume an, die sowohl die gleiche pre- als auch post-order Linearisierung besitzen. Minimal bedeutet hier mit der kleinstmöglichen Anzahl von Elementen wobei jedes Element maximal einmal pro Baum enthalten sein darf.

Aufgabe 3 ($\mathcal{O}$-Notation): (2+2+2+3+2 Punkte)

Zeigen oder widerlegen Sie die folgenden Aussagen:

a) $g(n) = 2n^3 + 142n^2 + 462 \in \Theta(n^3)$
b) $2^{n+1} \in \Theta(2^n)$
c) $\log n \in \mathcal{O}(\sqrt{n})$
d) $max(f(n),\ g(n)) \in \Theta(f(n) + g(n))$
e) $g(n) + f(n) \in \mathcal{O}(g(f(n)))$

Aufgabe 4 (Beweise): (3+3 Punkte)

Zeigen oder widerlegen Sie die folgenden Aussagen:

a) $o(f(n)) \cap \omega(f(n)) = \emptyset$
b) Aus $f(n) \in \Omega(g(n))$ und $g(n) \in \Omega(h(n))$ folgt $f \in \Omega(h(n))$

Aufgabe 5 (Laufzeitanalyse): (3 Punkte)

Bestimmen Sie die Komplexitätsklasse für den Aufruf `berechne(n)` in Abhängigkeit von n. Gehen Sie davon aus, dass die Grundrechenarten +, -, *, / in konstanter Zeit O(1) ausgeführt werden, ebenso die Zuweisungen = und Vergleiche >.

```
int calculate(int value){

    int result = value;

    for(int i = value; i > 0; i = i/2){
        result = result * result;
        for(int j = i; j > 0; j--){
            result = 2 * result;
        }
    }

    return result;
}
```