Ludwig-Maximilians-Universität München München, 27.04.2021
Institut für Informatik
Prof. Dr. Thomas Seidl
Andrea Maldonado, Florian Richter

Algorithmen und Datenstrukturen

SS 2021

Übungsblatt 2: Grundlagen und Komplexität

Tutorien: 03.05.-07.05.2021

Aufgabe 2-1 *Traversieren*

Gegeben ist das Ergebnis einer Preorder- und Inorder-Traversierung für ein und denselben binären Baum.

PreOrder-Traversierung = A E F D I H G
InOrder-Traversierung = F E A H I D G

Die Knotenbezeichnungen sollen in dieser Aufgabe immer eindeutig sein.

(a) Geben Sie einen Baum an, zu dem die PreOrder- und InOrder-Traversierung passt.

(b) Geben Sie zwei Binärbäume $T_1, T_2, T_1 \neq T_2$ an, für die gilt:
PreOrder(T_1) $=$ PreOrder(T_2) aber InOrder(T_1) $\neq$ InOrder(T_2)

(c) Gibt es zwei Binärbäume T_1, T_2 mit $T_1 \neq T_2$, für die gilt:
PreOrder(T_1) $=$ PreOrder(T_2) und InOrder(T_1) $=$ InOrder(T_2).
Begründen Sie Ihre Antwort. Ein formaler Beweis ist nicht notwendig.

Aufgabe 2-2 *O-Notation*

Zeigen oder widerlegen Sie:

(a) $O(n) \subseteq O(n^2)$

(b) $log_{10}(x) \notin O(log_2(x))$

(c) $O(n^2) \subseteq O(n * log_2(n))$

(d) $20000 * g(n) \in O(g(n))$

(e) $g(n) * n^2 \in O(n^2)$

Aufgabe 2-3 *Komplexitätsklassen*

Vergleichen Sie die Komplexitätsklassen $O(\log n)$, $O(\sqrt{n})$, $O(\log^2 n)$, $O(\log(n^2))$, $O(n)$, $O(log(log(n)))$ und $O(\log^k n)$ miteinander. Zeigen Sie die Korrektheit der von Ihnen gefundenen Ordnung.

Aufgabe 2-4 *Rekursionsgleichungen und Mastertheorem*

a) Der Algorithmus, um die Türme von Hanoi zu lösen, hat als Java-Code folgende Form:

```
public static void Hanoi(int Scheibenindex, int start, int ziel, int temp){

  if(Scheibenindex == 1) {
    VersetzeObersteScheibe(start, ziel);
  }

  else {
    Hanoi(Scheibenindex - 1, start, temp, ziel);
    VersetzeObersteScheibe(start, ziel);
    Hanoi(Scheibenindex - 1, temp, ziel, start);
  }
}
```

Die Zeitkomplexität der Funktion VersetzeObersteScheibe(...) sei $\mathcal{O}(1)$. Stelle die Rekursionsgleichung für diesen Algorithmus auf und verwende das Mastertheorem, um die Zeitkomplexität in Abhängigkeit des Parameters $Scheibenindex$ zu ermitteln.

b) Bestimme die Komplexitätsklasse für die folgenden Rekursionsgleichungen T(n) mithilfe des Mastertheorems. Geben Sie die entsprechenden Werte für alle Variablen der (unten gegebenen) Rekursionsungleichung an.

(i) $T(n) = 2T(\frac{n}{2}) + n^3$

(ii) $T(n) = 3 \cdot T(n-1) + n^3$

(iii) Wir definieren die Tribonacci-Zahlen: $T_0 = 1, T_1 = 1, T_{n+3} = T_{n+2} + T_{n+1} + T_n$.