# Übriggebliebene Aufgaben

(1) Der Binomialkoeffizient $\binom{n}{k}$ kann folgendermaßen rekursiv berechnet werden:

$$\binom{n}{k} = \begin{cases} 0, & \text{falls } k > n; \\ 1, & \text{falls } k = 0 \text{ oder } n = k; \\ \binom{n-1}{k-1} + \binom{n-1}{k}, & \text{sonst.} \end{cases}$$

Geben Sie ein Programm in Pseudocode an, welches $\binom{n}{k}$ mittels dynamischer Programmierung und Bottom-Up-Ansatz berechnet. Hinweis: Verwenden Sie eine Matrix (d.h. ein 2-dimensionales Array), um die Lösungen der Teilprobleme zu speichern.

(2) Zeigen oder widerlegen Sie: Ein kürzester-Wege-Baum ist auch ein minimaler Spannbaum.

(3) (a) Zeigen Sie: Dijkstra liefert bei negativen Kanten in gerichteten Graphen auch ohne negative Kreise im Allgemeinen ein falsches Ergebnis.

(b) Geben Sie einen stark zusammenhängenden, gerichteten Graphen mit mindestens einer negativen Kante an, so dass Dijkstra für mindestens einen Startknoten ein korrektes Ergebnis liefert.

(c) Warum funktioniert Dijkstra nicht auf zusammenhängenden, ungerichteten Graphen mit mindestens einer negativen Kante?

(4) Lösen Sie folgende Rekurrenzgleichungen:

(a) $T(n) = T(\frac{n}{2}) + n$

(b) $T(n) = 2 \cdot T(\frac{n}{2}) + 1$

(c) $T(n) = T(\frac{n}{2}) + 1$

(5) Warum lässt sich das Mastertheorem nicht auf folgende Gleichungen anwenden?

(a) $T(n) = T(\frac{n}{2}) + \log(n)$

(b) $T(n) = n \cdot T(\frac{n}{2}) + 1$

(c) $T(n) = T(\frac{n}{2}) + n \cdot (2 - \cos(n))$

(d) $T(n) = T(n) + 1$

(6) Seien $f, g, h, j : \mathbb{N} \to \mathbb{N}_0$ monoton wachsend. Zeigen oder widerlegen Sie:

(a) $f \in \Theta(g)$ und $h \in \Theta(j) \Rightarrow f \circ h \in \Theta(g \circ j)$

(b) $f \in \Theta(g)$ und $h \in \Theta(j) \Rightarrow f \cdot h \in \Theta(g \cdot j)$

(c) $f \in \Theta(g)$ und $h \in \omega(j) \Rightarrow f + h \in \Theta(g + j)$

(d) $f \in o(g) \Rightarrow h \circ f \in o(h \circ g)$

(e) $f \in O(g) \Rightarrow h \circ f \in O(h \circ g)$

(f) $f \in o(g) \Rightarrow h \circ f \in O(h \circ g)$

(g) $f \in O(f^2)$

(h) $f \in o(f^2)$

(i) $f \in \Theta(f^2)$

(7) Schreibe die Methode $bh(x, T)$, die für jeden Knoten $x$ seine Schwarzhöhe zurückgibt.

(8) Schreibe eine Methode $check\_b\_r\_tree(T)$, die einen Rot-Schwarz-Baum als Parameter nimmt und überprüft, ob dieser ein gültiger Rot-Schwarz-Baum ist.

(9) (a) Ein Rot-Schwarz-Baum habe die Höhe $h$. Berechne die maximale Differenz zwischen Entfernungen von der Wurzel zu den NIL-Knoten (mit kurzer Begründung).

(b) Konstruiere einen solchen Rot-Schwarz-Baum für $h = 3$.

(10) (a) Gegeben sei ein Rot-Schwarz-Baum der Höhe $h$ mit genau einem roten Knoten. Dieser habe den Abstand $d$ von der Wurzel. Berechne für jedes $n \in \mathbb{N}$ die Anzahl $a(n)$ der NIL-Knoten, die den Abstand $n$ zur Wurzel haben.

(b) Ist es möglich, dass $a(h) = 2^h$? Begründe!

(11) Was muss für eine Hashtabelle mit Verkettung gelten, damit der Aufwand für die Suche nach einem Element O(1) beträgt?

(12) (a) Was bedeutet der Begriff „amortisierte Analyse"?

(b) Nennen Sie ein Beispiel für eine Operation, die in der amortisierten Analyse eine bessere Laufzeit hat als im Worst Case. Geben Sie zusätzlich die beiden Laufzeiten an. Eine Begründung ist nicht notwendig.

(c) Nennen Sie eine Gemeinsamkeit und einen Unterschied zwischen amortisierter Analyse und Average-Case-Laufzeit.