

Übungsblatt 5

Ausgabe: 17.05.2022
Abgabe: 24.05.2022, **08:00**

Aufgabe 5.1 *Eigenschaften von AVL-Bäumen*

(2 + 5 + 5* Punkte)

- Beschreiben Sie, wie in Zeit $\mathcal{O}(\log n)$ das kleinste und das größte Element in einem AVL-Baum gefunden werden kann.
- Wir betrachten den Fall, dass in einem AVL-Baum ein neuer Schlüssel am Ende des Suchpfades als Blatt eingefügt wurde. Bei wie vielen Knoten muss im schlimmsten Fall die AVL-Eigenschaft wiederhergestellt werden? Geben Sie in Ihrer Begründung insbesondere die Tiefe aller relevanten Knoten und Teilbäume an.
- c*) Sei $\min(t)$ die minimale Knotenzahl, die ein AVL-Baum der Tiefe t haben muss. Zeigen Sie durch vollständige Induktion, dass

$$\min(t) = \frac{1}{\sqrt{5}} (\alpha^{t+3} - \beta^{t+3}) - 1$$

für alle $t \in \mathbb{N}_0$ gilt, wobei $\alpha = \frac{1+\sqrt{5}}{2}$ und $\beta = \frac{1-\sqrt{5}}{2}$. Sie können die Rekursionsgleichung für $\min(t)$ aus der Vorlesung ohne weitere Begründung verwenden.

Hinweis: Die Identitäten $1 + \alpha = \alpha^2$ und $1 + \beta = \beta^2$ können im Induktionsschritt hilfreich sein.

Aufgabe 5.2 *Binärbäume gesucht!*

(4 + 4 + 3 + 2 Punkte)

Die Kriminalabteilung der Agency for Lost Graph Objects (ALGO) bittet Sie um Ihre Mithilfe: Es wurden einige Binärbäume als vermisst gemeldet. Trotz zahlreicher Suchaktionen hat man bislang leider keinen der vermissten Binärbäume finden können. Im Zuge der Ermittlungen wurden lediglich Paare von Knotenfolgen gefunden, bei denen es sich laut fachkundiger Einschätzung um sauber ausgeführte Traversierungen handelt. Als gesichert gilt, dass die beiden Traversierungen in jedem Paar zu demselben Binärbaum gehören. Dabei wurde immer der Teilbaum des linken Kindes eines Knotens vor dem Teilbaum seines rechten Kindes betrachtet, auch wenn der linke Teilbaum leer war (beispielsweise wurde in diesem Fall also bei Inorder der Knoten selbst *vor* dem Teilbaum seines rechten Kindes betrachtet). Sie sollen nun prüfen, inwiefern sich aus den gefundenen Traversierungen Rückschlüsse auf die gesuchten Binärbäume ziehen lassen.

Rekonstruieren Sie für die Ihnen in a) und b) von der ALGO übermittelten Paare von Traversierungen jeweils den entsprechenden Binärbaum und erläutern Sie Ihr Vorgehen.

a) 35, 8, 10, 4, 23, 16, 19, 17, 28 (Präorder) und 10, 8, 4, 35, 19, 16, 17, 23, 28 (Inorder).

b) 11, 9, 3, 42, 8, 6, 12 (Postorder) und 11, 42, 9, 3, 12, 6, 8 (Inorder).

c) Aufgrund Ihrer erfolgreichen Arbeit übermittelt Ihnen die ALGO ein weiteres Paar:

5, 18, 7, 22, 21, 9, 6 (Postorder) und 6, 7, 5, 18, 9, 22, 21 (Präorder).

Aus nicht näher genannten Quellen weiß die ALGO bereits, dass es sich dabei um einen *vollen* Binärbaum handelt. Rekonstruieren Sie diesen Baum und erläutern Sie Ihr Vorgehen.

Hinweis: In einem vollen Binärbaum hat jeder Knoten entweder keine oder zwei Kinder.

d) Sie möchten die ALGO vor dem Trugschluss warnen, dass für *alle* Paare, die aus einer Postorder- sowie einer Präorder-Traversierung mit mehr als einem Knoten bestehen, der entsprechende Binärbaum eindeutig rekonstruiert werden kann. Zeigen Sie dies durch ein entsprechendes Gegenbeispiel.

Aufgabe 5.3 *Nahezu sortiert*

(6 Punkte)

Sei A ein Array mit $n > 0$ natürlichen Zahlen, wobei keine Zahl mehr als einmal vorkommt. Die Einträge in A seien bezüglich eines Parameters $k \leq n$ „nahezu“ sortiert, d.h., jeder Eintrag ist höchstens um k Positionen von seiner eigentlich korrekten Position in der Sortierung entfernt.

Entwerfen Sie einen Algorithmus, der die Einträge im Array A in die korrekte Sortierung überführt. Die Laufzeit sollte $\mathcal{O}(n \log k)$ nicht überschreiten. Beschreiben Sie wie immer zuerst die Idee Ihres Algorithmus und geben Sie ihn dann in Pseudocode an. Zeigen Sie, dass der Algorithmus korrekt arbeitet und die genannte Laufzeitschranke einhält.

