Lehrstuhl für Informatik I
Algorithmen, Komplexität und wissensbasierte Systeme
Universität Würzburg

Würzburg, den 5. Dezember 2019

Prof. Dr. Alexander Wolff
Andre Löffler, M. Sc.

7. Übungsblatt zur Vorlesung Algorithmen und Datenstrukturen (Winter 2019/20)

Aufgabe 1 – Höhe von Rot-Schwarz-Bäumen

Im Folgenden beziehen sich die Anzahl der Knoten und die Höhe eines Rot-Schwarz-Baums auf den Baum *ohne* die nil-Blätter.

Stellen Sie sicher, dass rote und schwarze Knoten klar voneinander unterscheidbar sind. Sie brauchen die nil-Blätter nicht zu zeichnen.

a) Zeichnen Sie einen gültigen Rot-Schwarz-Baum der Höhe 3, bei dem die Anzahl der Knoten minimal ist. **2 Punkte**

b) Zeichnen Sie einen binären Suchbaum der Höhe 3, für den es keine Färbung gibt, welche die Eigenschaften eines Rot-Schwarz-Baums erfüllt. Dieser Suchbaum soll die maximale Anzahl von Knoten enthalten. **2 Punkte**

Aufgabe 2 – Operationen in Rot-Schwarz-Bäumen

a) In einen anfangs leeren Rot-Schwarz-Baum werden nacheinander die Zahlen $50, 42, 38, 13, 17$ und 5 eingefügt. Zeichnen Sie den Baum nach jeder dieser Einfügeoperationen. **3 Punkte**

b) Anschließend werden die Zahlen $5, 13, 17$ und 38 nacheinander gelöscht. Zeichnen Sie den Baum nach jeder dieser Löschoperationen. **2 Punkte**

Stellen Sie sicher, dass rote und schwarze Knoten klar voneinander unterscheidbar sind. Sie brauchen die nil-Blätter nicht zu zeichnen.

Aufgabe 3 – Pfade im Suchbaum

Gegeben sei eine Folge ganzer Zahlen $A[1..k]$. Wir betrachten das Problem, zu entscheiden, ob es einen binären Suchbaum B gibt, so dass A die Folge von Zahlen ist, die bei der Suche nach $A[k]$ in B evaluiert wird. $A[1]$ wäre also die Wurzel von B, $A[2]$ ein Kind der Wurzel usw.

a) Lösen Sie dieses Entscheidungsproblem für die folgenden Zahlenfolgen. Begründen Sie jeweils Ihre Antwort. **3 Punkte**

1. $A = \langle 2, 252, 401, 398, 330, 344, 397, 363 \rangle$
2. $A = \langle 924, 220, 911, 244, 898, 258, 362, 363 \rangle$
3. $A = \langle 935, 278, 347, 621, 299, 392, 358, 363 \rangle$

b) Beschreiben Sie mit Worten und in Pseudocode einen Algorithmus, der dieses Entscheidungsproblem für eine beliebige gegebene Zahlenfolge löst. Die Worst-Case-Laufzeit des Algorithmus soll $\Theta(k)$ sein. **3 Punkte**

Aufgabe 4 – Listen-Augmentierung

Die Datenstruktur *doppelt verkettete Liste* soll um eine Methode Invert erweitert werden, nach deren Ausführung sich die Methoden der Liste so verhalten, als ob die Listenelemente in umgekehrter Reihenfolge in der Liste stehen würden, beispielsweise durchsucht Search die Liste nun von hinten nach vorne. Nach erneutem Aufruf von Invert soll die Reihenfolge wieder umgekehrt werden.

Beispiel: Gegeben sei die Liste $A = \langle 1, 2, 3 \rangle$. Führt man die Operation A.Insert(4) aus, so ergibt sich die Liste $\langle 4, 1, 2, 3 \rangle$. Nachdem man nun nacheinander die Operationen

A.Invert, A.Insert(5), A.Invert

ausgeführt hat, ergibt sich die Liste $\langle 4, 1, 2, 3, 5 \rangle$.

Skizzieren Sie in Worten, wie man die aus der Vorlesung bekannte Liste augmentieren kann, so dass Invert nur $O(1)$ Zeit benötigt und sich die asymptotische Worst-Case-Laufzeiten von Search, Insert und Delete nicht ändern. Erklären Sie, wie Sie Invert implementieren und wie Sie die genannten Operationen der Liste anpassen um die Anforderungen zu erfüllen. **5 Punkte**

Bitte werfen Sie Ihre Lösungen bis **Donnerstag, 12. Dezember 2019, 13:00 Uhr** in den Vorlesungs-Briefkasten im Informatik-Gebäude. Geben Sie stets die Namen und Übungsgruppen aller BearbeiterInnen sowie die Übungsgruppe, in der das Blatt zurückgegeben werden soll, an.

Grundsätzlich sind stets alle Ihrer Aussagen zu begründen und Ihr Pseudocode ist stets zu kommentieren.

Die Lösungen zu den mit PABS gekennzeichneten Aufgaben, geben Sie bitte nur über das PABS-System ab. Vermerken Sie auf Ihrem Übungsblatt, in welchem Repository (*sXXXXXX*-Nummer) die Abgabe zu finden ist. Geben Sie Ihre Namen hier als Kommentare in den Quelltextdateien an.