

## 6. Übungsblatt zur Vorlesung Algorithmen und Datenstrukturen (Winter 2017/18)

### 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 – Rot-Schwarz-Baum augmentieren

Ein Rot-Schwarz-Baum zur Verwaltung einer dynamischen Menge verschiedener ganzer Zahlen soll so augmentiert werden, dass man zu jeder Zeit bestimmen kann, für welche zwei Zahlen  $i, j$  der Menge mit  $i < j$  die Differenz  $j - i$  am kleinsten ist.

- a) Geben Sie die Methode `MinGap` in Pseudocode an, die das gesuchte Zahlenpaar in konstanter Zeit liefern soll.  
Benennen Sie die Extrainformation, die dafür zu speichern ist, und geben Sie an, wie Sie diese in den Methoden `Insert`, `Delete` und `Search` aufrechterhalten können, ohne deren asymptotische Worst-Case-Laufzeiten zu verschlechtern. **4 Punkte**
- b) Können Sie das Problem auch mit konstantem Speicher für Extrainformation lösen, wenn Sie auf die Methode `Delete` verzichten (also nur eine *halbdynamische* Menge verwalten)? **3 Punkte**

### Aufgabe 3 – Binär hochzählen

Die Datenstruktur  $D$  enthält eine einzige natürliche Zahl  $Z$  in Binärdarstellung (zu Beginn ist  $Z = 0$ ) und stellt lediglich die Methode `Increment` zur Verfügung, die  $Z$  um Eins erhöht. Die Laufzeit der Methode entspricht dabei der Anzahl der Bits, die sich in  $Z$  durch die Erhöhung um Eins ändern.

Zeigen Sie, dass die *amortisierte* Laufzeit von `Increment` in  $O(1)$  ist!

**4 Punkte**

### Aufgabe 4 – Spezialsuche

Gegeben sei ein Feld  $A[1..k]$  mit ganzen Zahlen, für die  $A[1] < A[2] < \dots < A[k]$  gilt. Geben Sie in Worten und im Pseudocode einen Algorithmus an, der ermittelt, ob es eine Zahl  $j \in \{1, \dots, k\}$  mit  $A[j] = j$  gibt. Die Worst-Case Laufzeit Ihres Algorithmus soll  $\Theta(\log k)$  sein.

**5 Punkte**

*Hinweis:* Finden Sie ein geeignetes, leicht zu berechnendes Kriterium, um damit den gesuchten Index zu suchen. Bedenken Sie auch die besondere Struktur von  $A$ .

---

Bitte werfen Sie Ihre Lösungen bis **Dienstag, 9. Januar 2018, 10:10 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.