Algorithmen und Programmieren 3, WS 2003/2004.
zweite Nachklausur zur ersten Teilklausur
Dienstag, 6. April 2004.

Anleitung: Streichen Sie *eine* Aufgabe deutlich auf dem Angabeblatt. Diese Aufgabe wird nicht in die Bewertung einbezogen. Wenn Sie selbst keine Aufgabe streichen, wird die erste Aufgabe nicht in die Bewertung einbezogen. Bearbeiten Sie die übrigen drei Aufgaben.

Bearbeiten Sie jede Aufgabe auf einem getrennten Blatt. Jede Aufgabe hat 10 Punkte. Bearbeitungszeit: 90 Minuten

---

1. *Rot-Schwarz-Bäume* sind binäre Bäume mit folgenden Eigenschaften:[1]

   (a) Jeder innere Knoten hat zwei Kinder.

   (b) Jeder Knoten ist entweder als *rot* oder als *schwarz* gekennzeichnet.

   (c) Die Wurzel und alle Blätter sind schwarz.

   (d) Die Kinder eines roten Knotens sind schwarz.

   (e) Ein schwarzer Knoten kann höchstens ein rotes Kind haben.

   (f) Alle Wege von der Wurzel zu den Blättern enthalten die gleiche Anzahl von schwarzen Knoten.

   Die *schwarze Höhe* ist die um eins verminderte Anzahl der schwarzen Knoten auf jedem Weg von der Wurzel zu einem Blatt.

   (a) Welche Höhe kann ein Baum mit schwarzer Höhe h' mindestens und höchstens haben?

   (b) Wie viele rote Knoten kann ein Baum mit schwarzer Höhe h' mindestens und höchstens haben?

   (c) Wie viele schwarze Knoten kann ein Baum mit schwarzer Höhe h' mindestens und höchstens haben?

   Begründen Sie Ihre Antworten.

2. Fügen Sie nacheinander die Werte 5, 3, 9, 2, 7, 10, 11, 4, 1 in einen anfangs leeren 2-3-Baum ein. Entfernen Sie dann die Element 5 und 7. Zeichnen Sie den 2-3-Baum nach jeder Operation auf.

3. Für eine gegebene Folge von n verschiedenen Zahlen $x_1, x_2, \dots, x_n$ soll die Anzahl der *Inversionen* (Fehlstände) bestimmt werden, das sind die Paare (x_i, x_j) mit $1 \leq i < j \leq n$ und $x_i > x_j$. Zum Beispiel hat die Folge $(3, 5, 10, 1, 6)$ vier Inversionen, nämlich die Paare $(3, 1)$, $(5, 1)$, $(10, 1)$ und $(10, 6)$.

   Zeigen Sie, wie man den Algorithmus *Sortieren durch Verschmelzen* so erweitern kann, dass er in $O(n \log n)$ Zeit die Anzahl der Inversionen berechnet. Geben Sie insbesondere genau an, was man beim Verschmelzen machen muss.

   Wahlweise dürfen Sie auch einen anderen Algorithmus angeben, der das Problem in $O(n \log n)$ Zeit löst.

4. Schreiben Sie ein Programm in Java oder Haskell, das eine sortierte Liste von n ganzen Zahlen in einen binären Suchbaum der kleinstmöglichen Höhe $h = \lceil \log_2(n+1) \rceil - 1$ umwandelt.

   (Ihr Programm muss *nicht* vollständig in dem Sinn sein, dass Sie zum Beispiel das Suchen und alle notwendigen und nützlichen Methoden für eine Klasse `Baum` programmieren müssen. Beschränken Sie sich auf die gestellte Aufgabe.)

---

[1]Diese Definition stimmt mit der Definition auf dem 6. Übungsblatt überein.