

**Aufgabenblatt 7**

**Abgabe: 20. Juni, 20.00 Uhr**

**1. Interpolationssuche auf Strings** (mittel) (schriftlich, 3+2 Punkte)

Interpolationssuche in einem Feld ist immer dann gut geeignet, wenn die Schlüssel annähernd gleichverteilt sind (wie z.B. in einem Lexikon oder Telefonbuch). Voraussetzung ist, dass man mit den Schlüsseln „rechnen“ kann. Sucht man mittels Interpolationssuche in einem Feld mit Zahlen, so ist dies einfach. Bei Strings gestaltet sich die Aufgabe etwas komplizierter.

a) Schreiben Sie eine Prozedur zur Interpolationssuche in einem Feld von Strings. Die Strings sollen dabei lexikographisch (d.h. wie im Telefonbuch) sortiert sein.

b) Testen Sie Ihre Prozedur auf zufällig erzeugten Daten. Zur Sortierung können Sie einen der beiden Sortieralgorithmen von Aufgabenblatt 1 oder 3 oder das Quicksortverfahren aus der Vorlesung verwenden.

Die Strings sollen jeweils eine zufällige Länge zwischen 1 und 10 haben, es sollen dabei nur die Kleinbuchstaben (a,...,z) verwendet werden. Testen Sie ihre Prozedur auf Feldern mit 100, 1000 und 10000 Strings. Suchen Sie jeweils nach 100 Zufallsstrings und ermitteln Sie die mittlere Anzahl der Vergleiche, bis das Element gefunden wurde (bei Zufallsstrings ist zu erwarten, dass die Suche erfolglos abbricht). Suchen Sie jeweils nach 100 zufällig aus dem Feld entnommenen Strings und ermitteln Sie die mittlere Anzahl der Vergleiche, bis das Element (erfolgreich) gefunden wurde.

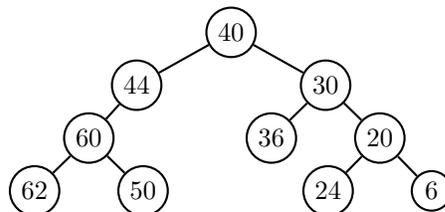
**2. Eigenschaften von Bäumen** (leicht) (votieren, 1 Punkt)

Was ist die minimale und die maximale Anzahl von Knoten, die ein  $k$ -ärer Baum der Höhe  $h$  haben kann?

**3. Baumtraversierungen** (leicht) (votieren, 1.5+1.5+0.5 Punkte)

Wiederholen Sie die Baumtraversierungen aus Abschnitt 1.6.4.4

a) Geben Sie Pre-, In- und Postorder des folgenden Baums an.



b) Ist der obige Baum durch die Pre- und Postordertraversierung eindeutig bestimmt? Geben Sie einen weiteren Baum mit denselben Pre- und Postordertraversierungen an oder begründen Sie, warum ein solcher nicht existieren kann.

c) Geben Sie ein Traversierungsschema an, das die Knoten eines Suchbaums (Definition siehe unten oder Skript) in absteigend sortierter Reihenfolge ausgibt.

**4. Bäume sind 2-färbbar** (leicht) (votieren, 2 Punkte)

Ein beliebiger Graph  $G = (V, E)$  heißt  $k$ -färbbar, wenn eine Abbildung  $f : V \rightarrow \{1, 2, \dots, k\}$  existiert mit  $f(x) \neq f(y)$  für alle Kanten  $\{x, y\}$  bzw.  $(x, y)$ .

Die Frage, ob ein beliebiger Graph  $k$ -färbbar ist, i.A. nur mit großem Zeitaufwand lösbar. Bäume sind hingegen stets 2-färbbar. Geben Sie einen Algorithmus an, der für beliebige Bäume eine 2-Färbung berechnet.

**5. Suchbäume** (mittel) (schriftlich, 2.5+2+2+2 Punkte)

Ein binärer Baum, dessen Inhalts-Datentyp geordnet ist (z.B. ganze Zahlen), heißt (binärer) Suchbaum, wenn für alle Knoten  $u$  gilt: Alle Inhalte von Knoten im linken Unterbaum von  $u$  sind echt kleiner als der Inhalt von  $u$  und alle Inhalte im rechten Unterbaum von  $u$  sind größer oder gleich dem Inhalt von  $u$ .

In dieser Aufgabe sollen die Elemente im Suchbaum stets alle paarweise verschieden sein (d.h., auch im rechten Unterbaum von  $u$  sind alle Inhalte echt größer dem Inhalt von  $u$ ).

Entwerfen Sie möglichst effiziente Algorithmen, um folgende Operationen auf Suchbäumen zu realisieren. Geben Sie jeweils den Aufwand in  $O$ -Notation an.

- a) Finden, Einfügen und Löschen eines Elementes im Suchbaum (Algorithmus „Finden“ ist schon im Skript angegeben; ist ein Element schon im Suchbaum, so soll es bei „Insert“ nicht noch einmal eingefügt werden).
- b) Vereinigung zweier Suchbäume: Gesucht ist ein Suchbaum, der alle Elemente enthält, die in mindestens einem der beiden Suchbäume vorhanden sind.
- c) Schnitt zweier Suchbäume: Gesucht ist ein Suchbaum, der alle Elemente enthält, die in beiden Suchbäumen vorhanden sind.
- d) Gleichheit: Gesucht ist eine Prozedur, die testet, ob zwei Suchbäume dieselben Elemente enthalten.

Alles weitere unter

[http://www.informatik.uni-stuttgart.de/fmi/fk/lehre/ss03/info\\_II/default.htm](http://www.informatik.uni-stuttgart.de/fmi/fk/lehre/ss03/info_II/default.htm)

Fragen zur Vorlesung und den Übungen, sowie Anregungen und Kritik können auf dem Schwarzen Brett

<http://fachschaft.informatik.uni-stuttgart.de/forum/>

diskutiert werden.