

## 9. Übungsblatt

zur Vorlesung

# Grundzüge der Informatik I

Abgabe über Ilias bis zum 14.6. 14:00 Uhr.  
Besprechung in Kalenderwoche 25.

### Aufgabe 1 Binärbäume (3 + 2 + 2 + 3 + 2 Punkte)

- a) Fügen Sie die Schlüssel 31, 1, 5, 87, 65, 19 in dieser Reihenfolge in einen anfangs leeren binären Suchbaum ein.
- b) Übersetzen Sie Ihren Suchbaum aus Aufgabenteil a) in die Darstellung im Speicher eines Rechners. Nehmen Sie dazu folgendes an:

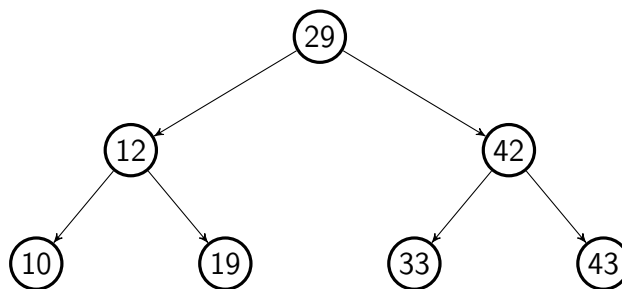
1. Die Knoten des Baums liegen, beginnend bei Speicheradresse 1, in der gleichen Reihenfolge aus a) im Speicher.
2. Der Baum liegt in einer ununterbrochenen Sequenz im Speicher (Zwischen den Speicherzellen für die Knoten sind keine anderen Speicherzellen).

- c) Betrachten Sie die folgende Sequenz an Speicherzellen.

12	0	13	5	31	1	9	0	24	5	0	0	9	1	17	0	17	13	0	0
----	---	----	---	----	---	---	---	----	---	---	---	---	---	----	---	----	----	---	---

Übersetzen Sie die gegebene Sequenz in einen Binärbaum und zeichnen Sie diesen.

- d) Betrachten Sie den folgenden Suchbaum.



Löschen Sie die Schlüssel 33, 42, 29 in dieser Reihenfolge aus dem Baum.

- e) Betrachten Sie den folgenden Algorithmus:

Traversierung( $x$ ):

```
1 if  $x \neq NIL$  then  
2   Traversierung(right[ $x$ ])  
3   Traversierung(left[ $x$ ])  
4   Ausgabe key[ $x$ ]
```

Wenden Sie, auf Ihrem Binärbaum aus Teilaufgabe c), den Algorithmus Traversierung an, und verwenden Sie den Wurzelknoten Ihres Binärbaums als initialen Eingabeparameter. Geben Sie die Ausgaben und deren Reihenfolge an.

**Aufgabe 2** *Datenstruktur entwickeln (8 Punkte)*

Sei  $k \in \mathbb{N}$  eine Konstante und  $U = \{1, \dots, k\}$  ein Universum. Gesucht ist eine Datenstruktur zur Repräsentation von Multimengen aus dem Universum  $U$ . Eine Multimenge  $M$  aus dem Universum  $U$  kann jedes Element  $x \in U$  gar nicht, einmal oder auch mehrmals enthalten. Die gesuchte Datenstruktur soll folgende Operationen in den jeweils angegebenen Laufzeiten unterstützen.

- $\text{Add}(M, x)$  fügt der Multimenge  $M$  das Element  $x$  einmal hinzu. Laufzeit  $\mathcal{O}(1)$ .
- $\text{Remove}(M, x)$  entfernt das Element  $x$  aus  $M$ , sodass  $x$  anschließend gar nicht mehr in  $M$  vorkommt. Laufzeit  $\mathcal{O}(1)$ .
- $\text{Count}(M, x)$  liefert, wie häufig  $x$  in  $M$  vorkommt. Laufzeit  $\mathcal{O}(1)$ .
- $\text{Merge}(M, N)$  liefert eine neue Multimenge  $R$ , sodass für jedes  $x \in U$  die Anzahl von  $x$  in  $R$  der Summe der Anzahlen von  $x$  in  $M$  und in  $N$  gleicht. Laufzeit  $\mathcal{O}(k)$ , also auch  $\mathcal{O}(1)$ , da  $k$  eine Konstante ist.

Beschreiben Sie in wenigen kurzen Sätzen wie Ihre Datenstruktur aufgebaut ist. Schreiben Sie für jede der geforderten Operationen kommentierten Pseudocode und analysieren Sie dessen asymptotische Worst-Case-Laufzeit. Begründen Sie kurz, dass die Datenstruktur korrekt arbeitet.