Binärer Suchbaum

Weiterführende Literatur:

- *Wikipedia-Artikel „Binärbaum“*
- *Wikipedia-Artikel „Binärer Suchbaum“*
- Saake und Sattler, *Algorithmen und Datenstrukturen*, Kapitel 14.2.1, Seite 348-358 (PDF 364-374)
- *Qualifizierungsmaßnahme Informatik: Algorithmen und Datenstrukturen 5*, Seite 4-14

Visualisierungstools

- `http://btv.melezinek.cz/binary-search-tree.html` Auch für Postorder etc

- `https://visualgo.net/bn/bst`

Ein binärer Suchbaum - häufig abgekürzt als **BST** (von englisch *Binary Search Tree*) - ist ein binärer Baum, bei dem die Knoten *„Schlüssel“* tragen, und die Schlüssel des *linken Teilbaums* eines Knotens nur *kleiner* (*oder gleich*) und die des *rechten* Teilbaums nur *größer*[1] als der Schlüssel des Knotens selbst sind.[2]

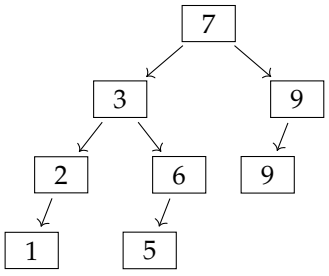
linken Teilbaums
kleiner (oder gleich)
rechten
größer

S	Wert eines Knotens
S_{links}	Wert der Wurzel des linken Teilbaumes
S_{rechts}	Wert der Wurzel des rechten Teilbaumes

Bei einem binären Suchbaum gilt: $S_{\text{links}} \leq S < S_{\text{rechts}}$

Beispiel

Anordnung von `7, 3, 6, 9, 2, 9, 5, 1` in einem int-Baum[3]



[1] *Qualifizierungsmaßnahme Informatik: Algorithmen und Datenstrukturen 5*, Seite 5.
[2] *Wikipedia-Artikel „Binärer Suchbaum“*.
[3] *Qualifizierungsmaßnahme Informatik: Algorithmen und Datenstrukturen 5*, Seite 5.

Traversierung eines Baums

Es gibt verschiedene Möglichkeiten einen binären Baum zu traversieren:

- `preorder`:

  besuche die Wurzel, dann den linken Unterbaum, dann den rechten Unterbaum; auch: WLR

```
62    private void besuchePreorder(BaumKnoten knoten, ArrayList<Comparable>
      ↪  schlüssel) {
63      if (knoten != null) {
64        schlüssel.add((Comparable) knoten.gibSchlüssel());
65        besuchePreorder(knoten.gibLinks(), schlüssel);
66        besuchePreorder(knoten.gibRechts(), schlüssel);
67      }
68    }
```

Code-Beispiel auf Github ansehen: `src/main/java/org/bschlangaul/baum/BinaerBaum.java`

- `inorder`:

  besuche den linken Unterbaum, dann die Wurzel, dann den rechten Unterbaum; auch: LWR

```
46    private void besucheInorder(BaumKnoten knoten, ArrayList<Comparable>
      ↪  schlüssel) {
47      if (knoten != null) {
48        besucheInorder(knoten.gibLinks(), schlüssel);
49        schlüssel.add((Comparable) knoten.gibSchlüssel());
50        besucheInorder(knoten.gibRechts(), schlüssel);
51      }
52    }
```

Code-Beispiel auf Github ansehen: `src/main/java/org/bschlangaul/baum/BinaerBaum.java`

- `postorder`:

  besuche den linken Unterbaum, dann den rechten, dann die Wurzel; auch: LRW

```
78    private void besuchePostorder(BaumKnoten knoten, ArrayList<Comparable>
      ↪  schlüssel) {
79      if (knoten != null) {
80        besuchePostorder(knoten.gibLinks(), schlüssel);
81        besuchePostorder(knoten.gibRechts(), schlüssel);
82        schlüssel.add((Comparable) knoten.gibSchlüssel());
83      }
84    }
```

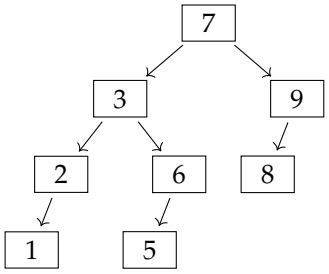
Code-Beispiel auf Github ansehen: `src/main/java/org/bschlangaul/baum/BinaerBaum.java`

- `levelorder`:

  Beginnend bei der Baumwurzel werden die Ebenen von links nach rechts durchlaufen.

```
97    private void besucheLevelorder(Warteschlange warteschlange,
      ↪  ArrayList<Comparable> schlüssel)
98        throws WarteschlangeFehler {
99      while (!warteschlange.istLeer()) {
100       BaumKnoten knoten = (BaumKnoten) warteschlange.verlasse();
101       if (knoten.gibLinks() != null)
102         warteschlange.betrete(knoten.gibLinks());
103       if (knoten.gibRechts() != null)
104         warteschlange.betrete(knoten.gibRechts());
105       schlüssel.add((Comparable) knoten.gibSchlüssel());
106     }
107   }
```

Code-Beispiel auf Github ansehen: `src/main/java/org/bschlangaul/baum/BinaerBaum.java`




```
preorder    7 3 2 1 6 5 9 8
inorder     1 2 3 5 6 7 8 9
postorder   1 2 5 6 3 8 9 7
levelorder  7 3 9 2 6 8 1 5
```

Löschen eines Knotens

Die komplizierteste Operation im binären Suchbaum ist das Löschen. Dies liegt daran, dass beim Entfernen eines inneren Knotens einer der beiden Teilbäume des Knotens „hochgezogen“ und gegebenenfalls umgeordnet werden muss.

Grundsätzlich müssen beim Löschen eines Knotens drei Fälle berücksichtigt werden:

(a) Der Knoten ist ein *Blatt*. Dieser Fall ist der einfachste, da hier nur der Elternknoten zu bestimmen ist und dessen Verweis auf den Knoten entfernt werden muss.

Blatt

(b) Der Knoten besitzt nur *einen Kindknoten*. In diesem Fall ist der Verweis vom Elternknoten auf den Kindknoten des zulöschenden Knotens umzulenken.

einen Kindknoten

(c) Der Knoten ist ein innerer Knoten mit *zwei Kindknoten*. Hierbei muss der Knoten durch den am *weitesten links stehenden Knoten des rechten Teilbaumes ersetzt* werden, da dieser in der Sortierreihenfolge der nächste Knoten ist. Alternativ kann auch der am weitestens rechts stehende Knoten des linken Teilbaumes verwendet werden.[4]

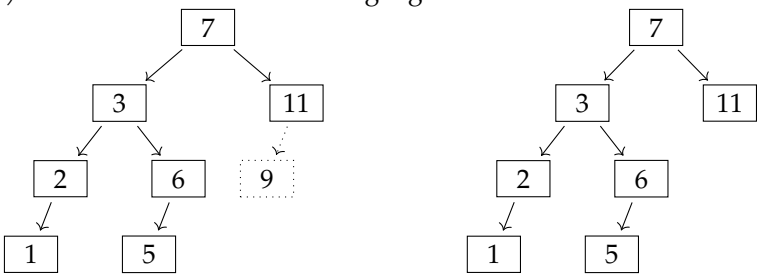
zwei Kindknoten

weitesten links stehenden Knoten des rechten Teilbaumes ersetzt

[4] Saake und Sattler, *Algorithmen und Datenstrukturen*, S. 364.

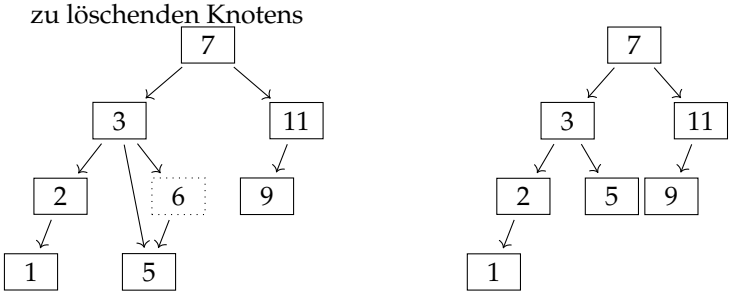
Knoten ohne Nachfolger[5]

(a) Suche den zu löschenden Knoten
(b) Lösche die Referenz vom Vorgängerkonten auf den zu löschenden Knoten



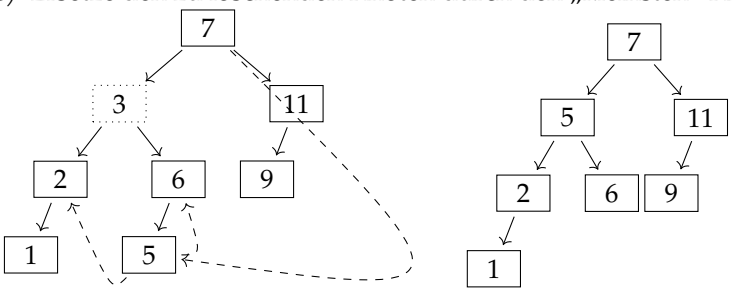
Knoten mit genau einem Nachfolger[6]

(a) Suche den zu löschenden Knoten
(b) Referenz von Vorgängerkonten auf den (einzigen) Nachfolgerknoten des zu löschenden Knotens



Knoten mit zwei Nachfolgern[7]

(a) Suche den zu löschenden Knoten
(b) Suche den „kleinsten“ Knoten im rechten Teilbaum
(c) Ersetze den zu löschenden Knoten durch den „kleinsten“ Knoten



[5] *Qualifizierungsmaßnahme Informatik: Algorithmen und Datenstrukturen 5*, Seite 10.
[6] *Qualifizierungsmaßnahme Informatik: Algorithmen und Datenstrukturen 5*, Seite 11.
[7] *Qualifizierungsmaßnahme Informatik: Algorithmen und Datenstrukturen 5*, Seite 12.

Literatur

[1] *Qualifizierungsmaßnahme Informatik: Algorithmen und Datenstrukturen 5. Bäume, Hashing*. https://www.studon.fau.de/file2619756_download.html.

[2] Gunter Saake und Kai-Uwe Sattler. *Algorithmen und Datenstrukturen. Eine Einführung in Java*. 2014.

[3] *Wikipedia-Artikel „Binärbaum“*. https://de.wikipedia.org/wiki/Binärbaum.

[4] *Wikipedia-Artikel „Binärer Suchbaum“*. https://de.wikipedia.org/wiki/Binärer_Suchbaum.