

Lernzettel

Binäre Suchbäume und Rotationen

Universität: Technische Universität Berlin
Kurs/Modul: Einführung in die Informatik - Vertiefung
Erstellungsdatum: September 20, 2025



Zielorientierte Lerninhalte, kostenlos!
Entdecke zugeschnittene Materialien für deine Kurse:

<https://study.AllWeCanLearn.com>

Einführung in die Informatik - Vertiefung

Lernzettel: Binäre Suchbäume und Rotationen

(1) Definition.

Ein binärer Suchbaum (BST) ist ein Binärbaum, bei dem für jeden Knoten K gilt: Alle Schlüssel im linken Teilbaum von K sind kleiner als der Schlüssel von K und alle Schlüssel im rechten Teilbaum von K sind größer als der Schlüssel von K . Duplikate werden je nach Implementierung unterschiedlich behandelt (z. B. Gleichheit erlauben und rechts einordnen).

(2) Eigenschaften.

- Invariante: Für jeden Knoten K gilt: $(L(K)) < (K) < (R(K))$, wobei $L(K)$ bzw. $R(K)$ die linken bzw. rechten Unterbäume von K sind.
- Operationen: Suchen, Einfügen, Traversierungen (Inorder, Preorder, Postorder).
- Komplexität (durchschnittlich): Such- und Einfügeoperationen $\mathcal{O}(\log n)$ bei balanciertem Baum; im Worstcase $\mathcal{O}(n)$ bei schiefem Baum.
- Traversierungen liefern eine bestimmte Reihenfolge der Schlüssel (z. B. Inorder liefert aufsteigend sortierte Folge).

(3) Grundoperationen im BST.

Suchen. Start am Wurzelknoten; falls der Schlüssel gleich ist, gefunden; ansonsten links oder rechts weiterlaufen, bis der Schlüssel gefunden wird oder ein leerer Knoten erreicht ist.

Einfügen.

Falls der Baum leer ist, wird der neue Schlüssel zum Wurzelknoten.

Ansonsten vom Wurzelknoten aus entsprechend der Vergleiche nach links oder rechts wandern und dort als Blatt einfügen.

Ziel ist es, die BST-Eigenschaft zu erhalten.

Traversierungen.

Inorder (Links—Knoten—Rechts): liefert eine sortierte Folge der Schlüssel.

Preorder (Knoten—Links—Rechts) und *Postorder* (Links—Rechts—Knoten) dienen u. a. Baumspeicher- und Kopieroperationen.

(4) Rotationen.

Rotationen dienen dazu, die Struktur eines BST zu verändern, ohne die BST-Eigenschaft zu verletzen. Sie werden in balancierten Varianten wie AVL- oder Rot-Schwarz-Bäumen verwendet, um die Tiefe des Baums zu begrenzen.

Rechtsrotation (RR) am Knoten y mit linkem Kind x :

Voraussetzung: $y.\text{left} \neq \emptyset$.

Ablauf (vereinfacht): 1) $T2 \leftarrow x.\text{right}$

2) $x.\text{right} \leftarrow y$

3) $y.\text{left} \leftarrow T2$

Hinweis: Falls y einen Elternknoten hat, muss dieser auf x verweisen; ist y die Wurzel des Subbaums, wird x neuer Wurzel des Subbaums.

Linksrotation (LR) am Knoten x mit rechtem Kind y :

Voraussetzung: $x.\text{right} \neq \emptyset$.

Ablauf (vereinfacht): 1) $T2 \leftarrow y.\text{left}$

2) $y.\text{left} \leftarrow x$

3) $x.\text{right} \leftarrow T2$

Hinweis: Wie oben, Elternverweis entsprechend aktualisieren.

Doppelte Rotationen.

- *LR* (Linksrotation der linken Teilbaums gefolgt von Rechtsrotation am Knoten): Zuerst Linksrotation im linken Unterbaum, dann Rechtsrotation am Knoten.
- *RL* (Rechtsrotation der rechten Teilbaums gefolgt von Linksrotation am Knoten): Zuerst Rechtsrotation im rechten Unterbaum, dann Linksrotation am Knoten.

(5) Beispiel (einfache Veranschaulichung).

Vorheriger Zustand:

```
  30
 /
20
 /
10
```

Rechtsrotation am Knoten 30 ergibt:

```
  20
 /  \
10   30
```

Damit hat sich die Struktur verändert, der BST-Constraint bleibt erhalten.

(6) Balancierung und Praxis.

- In einem reinen BST liegt der Schwerpunkt auf Korrektheit der BST-Eigenschaft; Balancierung erfolgt ggf. durch Zusatzstrukturen (AVL, Red-Black).
- Rotationen erhöhen die Balance und beeinflussen die Höhe des Baums, wodurch sich typische Laufzeitkomplexitäten verbessern können.

(7) Zusammenfassung.

- BST ist eine strukturierte Form von Binärbaum, der Such-, Einfüge- und Traversierungsoperationen effizient unterstützt.
- Rotationen verändern die Baumstruktur durch Umsortierung von Knoten, ohne BST-Eigenschaft zu verletzen; sie sind zentral in balancierten Varianten.