

# Baumstrukturen

Reiner Hüchting

22. März 2023

# Themenüberblick

Binärbäume

Binäre Suchbäume

Heaps

Präfixbäume

# Themenüberblick

Binärbäume

Binäre Suchbäume

Heaps

Präfixbäume

# Binärbäume

## Definition: Binärbaum

Für jeden Binärbaum gilt eine der folgenden Möglichkeiten:

- ▶ Der Baum ist leer.
- ▶ Der Baum besteht aus einer Wurzel und zwei Teil**bäumen**.

## Bemerkungen

- ▶ Rekursive Definition beschreibt direkt die Struktur.
- ▶ Verallgemeinerung zu *Bäumen* möglich.
- ▶ Häufig vorkommende Struktur in Mathematik und Informatik.

# Binärbäume

## Beispiele

- ▶ Turnierbäume bei Wettbewerben
- ▶ Wahrscheinlichkeitsbäume
- ▶ Stammbäume
- ▶ Modellierung von Abhängigkeiten

## Anwendung in der Informatik

- ▶ Strukturierung und Sortierung von Daten

# Themenüberblick

Binärbäume

Binäre Suchbäume

Heaps

Präfixbäume

# Themenüberblick

Binärbäume

Binäre Suchbäume

Binäre Suchbäume

AVL-Bäume

Heaps

Präfixbäume

# Binäre Suchbäume – Binäre Suchbäume

## Definition: Binärer Suchbaum

Ein binärer Suchbaum ist ein Binärbaum mit folgenden Eigenschaften:

- ▶ Die Knoten enthalten *Schlüssel* und *Werte* (engl. *key/value*).
- ▶ Auf den Schlüsseln ist eine *totale Ordnung* definiert.
- ▶ Für jeden Knoten gilt:
  - ▶ Die Schlüssel aller Knoten im linken Teilbaum sind kleiner als der Schlüssel der Wurzel.
  - ▶ Die Schlüssel aller Knoten im rechten Teilbaum sind größer als der Schlüssel der Wurzel.
  - ▶ (Gleichheit muss ggf. einer der Seiten zugeschlagen werden.)

## Ziel: Effiziente Implementierung von Listen und Datenbanken

- ▶ Idee: Binärer Suche und effiziente Sortierverfahren direkt in einer Datentstruktur verankern.



# Binäre Suchbäume – Binäre Suchbäume

## Auffinden von Elementen mit einem bestimmten Suchschlüssel

- ▶ Baum leer: Nicht gefunden.
- ▶ Suchschlüssel in Wurzel gefunden, liefere (Wert der) Wurzel.
- ▶ Suchschlüssel kleiner als Wurzel: Suche im linken Teilbaum.
- ▶ Suchschlüssel größer als Wurzel: Suche im rechten Teilbaum.

## Einfügen von Elementen mit einem bestimmten Suchschlüssel

- ▶ Baum leer: Hier einfügen.
- ▶ Suchschlüssel kleiner Wurzel: Füge in linken Teilbaum ein.
- ▶ Suchschlüssel größer Wurzel: Füge in rechten Teilbaum ein.

# Binäre Suchbäume – Binäre Suchbäume

## Löschen von Elementen mit einem bestimmten Suchschlüssel

- ▶ Suche das zu löschende Element.
- ▶ Falls Blatt: Entfernen.
- ▶ Ansonsten: Suche den direkten Nachfolger oder Vorgänger.
- ▶ Vertausche Element mit Nachfolger/Vorgänger.
- ▶ Entferne Element aus entsprechendem Teilbaum.

# Binäre Suchbäume – Binäre Suchbäume

## Verhalten im Optimalfall

- ▶ Linker und rechter Teilbaum in allen Knoten gleich tief.
- ▶ Logarithmisches Verhalten beim Suchen, Einfügen und Löschen.

## Verhalten im Worst Case

- ▶ Eine Seite hat starkes Übergewicht.
- ▶ Extremfall: Jeder Knoten hat nur einen Teilbaum.
- ▶ Der Baum ist dann de facto eine Liste.

# Themenüberblick

Binärbäume

Binäre Suchbäume

Binäre Suchbäume

AVL-Bäume

Heaps

Präfixbäume

# Binäre Suchbäume – AVL-Bäume

## Ziel: Optimierung von binären Suchbäumen

- ▶ Problem: Bäume können aus der Balance geraten.
- ▶ Lösung: Reorganisieren, wenn das Ungleichgewicht zu groß wird.

## Vorgehensweise

- ▶ Beim Einfügen oder Löschen Struktur des Baumes analysieren.
- ▶ Bei Bedarf Knoten umsortieren, damit linker und rechter Teilbaum jedes Knotens ungefähr gleich groß sind.
- ▶ Problem: Analyse darf nicht zu teuer sein.
- ▶ Frage: Was bedeutet „ungefähr gleich groß“ überhaupt?

# Binäre Suchbäume – AVL-Bäume

## Definition: Tiefe

Die Tiefe eines Knotens ist die Länge des Pfades von der Wurzel bis zu diesem Knoten.

## Bemerkungen

- ▶ Berechnung rekursiv:
  - ▶ Tiefe der Wurzel: 0.
  - ▶ Tiefe eines Knotens:  $1 + \text{Tiefe des Elternknotens}$
- ▶ Kann beim Einfügen/Löschen nebenbei berechnet werden.
- ▶ Kann bei Bedarf im Knoten gespeichert und gepflegt werden.

# Binäre Suchbäume – AVL-Bäume

## Definition: Höhe

Die Höhe eines Baums ist die Tiefe des tiefsten Knotens im Baum.

## Bemerkungen

- ▶ Berechnung rekursiv:
  - ▶ Höhe eines Blatts Wurzel: 1.
  - ▶ Höhe eines Knotens:  $1 + \text{Höhe des höheren Kindes}$
- ▶ Kann beim Einfügen/Löschen nebenbei berechnet werden.
- ▶ Kann bei Bedarf im Knoten gespeichert und gepflegt werden.

# Binäre Suchbäume – AVL-Bäume

## Definition: Balancefaktor

Der Balancefaktor eines Knotens ist die Differenz zwischen der Höhe des rechten und des linken Teilbaums.

## Bemerkungen

- ▶ Kann beim Einfügen/Löschen nebenbei berechnet werden.
- ▶ Gutes Maß für die Ausgeglichenheit des Baumes.



# Binäre Suchbäume – AVL-Bäume

## Definition: AVL-Baum

Ein AVL-Baum ist ein binärer Suchbaum mit folgender Eigenschaft:

- ▶ Der Balancefaktor jedes Knotens liegt im Intervall  $[-1, 1]$ .

## Umsetzung

- ▶ Beim Einfügen/Löschen Balancefaktoren bestimmen.
- ▶ Nach Einfügen in Teilbaum: Umorganisieren der Knoten.
- ▶ Nach Einfügen in Teilbaum bedeutet: Nach Rekursion.

# Binäre Suchbäume – AVL-Bäume

## Analyse der Balancefaktoren

Nach Einfügen eines Knotens Balancefaktor prüfen.

- ▶ Falls  $-2$ : Linker Teilbaum zu hoch.
- ▶ Falls  $2$ : Rechter Teilbaum zu hoch.
- ▶ Prüfe Balancefaktor des linken/rechten Teilbaumes.
- ▶ Führe passende *Rotation* durch.

## Ungleichgewichts-Situationen und Rotationen

- ▶ Links-Links
- ▶ Links-Rechts
- ▶ Rechts-Rechts
- ▶ Rechts-Links

# Binäre Suchbäume – AVL-Bäume

## Verhalten beim Einfügen/Löschen/Suchen

- ▶ Linker und rechter Teilbaum in allen Knoten fast gleich tief.
- ▶ Logarithmisches Verhalten beim Suchen, Einfügen und Löschen.

# Themenüberblick

Binärbäume

Binäre Suchbäume

Heaps

Präfixbäume

# Heaps

## Erinnerung: Binäre Suchbäume

- ▶ Idee: Optimales Such- und Einfügeverhalten sortierter Listen.
- ▶ Komplexität: Logarithmisches Verhalten wie bei binärer Suche.

Es geht besser, wenn keine perfekte Sortierung benötigt wird!

## Idee: Fordere nur eine partielle Sortierung

- ▶ Knoten müssen größer oder kleiner als ihre Kinder sein.
- ▶ Keine Relation zwischen den Kindern.
- ▶ Beobachtung: Größtes/Kleinstes Element steht an der Wurzel.
- ▶ Ermöglicht sehr schnellen Zugriff auf Wurzel.

# Heaps

## Definition: Vollständiger Binärbaum

Ein vollständiger Binärbaum ist ein Binärbaum mit folgenden Eigenschaften:

- ▶ Jede Ebene ist vollständig besetzt.
- ▶ Nur in der untersten Ebene dürfen Elemente fehlen.
- ▶ Ebenen werden beim Einfügen von links nach rechts aufgefüllt.

## Bemerkungen

- ▶ Ein vollständiger Binärbaum hat keine Lücken.
- ▶ Kann deshalb effizient als Liste gespeichert werden.
- ▶ Einfache Berechnung der Indizes:
  - ▶ Elternknoten an Stelle  $n$
  - ▶ Kinder an Stellen  $2n + 1$  und  $2n + 2$

# Heaps

## Definition: Min-Heap

Ein Min-Heap ist ein vollst. Binärbaum mit folgender Eigenschaft:

- ▶ Jeder Knoten ist kleiner als seine Kinder
- ▶ (Definition *Max-Heap* analog.)

## Einfügen von Elementen

- ▶ Neues Element am Ende einfügen.
- ▶ Aufsteigen lassen, bis es richtig eingeordnet ist.

## Entfernen der Wurzel

- ▶ Wurzel durch letztes Element ersetzen (tauschen).
- ▶ Absteigen lassen, bis es richtig eingeordnet ist.

# Heaps

## Verhalten beim Einfügen/Löschen von Elementen

- ▶ Zugriff auf Wurzel in konstanter Zeit ( $O(1)$ ).
- ▶ Einfügen und Löschen in  $O(\log n)$ .

## Anwendungen

- ▶ *Priority Queues*
- ▶ Routingverfahren, z.B. Navigationssysteme, Netzwerke
- ▶ Optimierungs- und Planungsprobleme
- ▶ effiziente Sortierverfahren (HeapSort)



# Themenüberblick

Binärbäume

Binäre Suchbäume

Heaps

Präfixbäume

# Präfixbäume

## Präfixbäume: Effiziente Speicherung von Zeichenketten

- ▶ Speichere Zeichenketten in einer Baumstruktur ab.
- ▶ Annotiere Knoten mit Eigenschaften dieser Zeichenketten.
- ▶ Zeichenketten mit gemeinsamem Präfix haben gemeinsame Pfade im Baum.
- ▶ Vorteile: Kompression und schnelle Suche.

## Anwendungen

- ▶ Metadaten von Textdokumenten
- ▶ Aufbau eines Suchindex für Texte
- ▶ Aufbau von Wörterbüchern (z.B. für *Predictive Text*)
- ▶ Kompressionsverfahren