



Imperative Programmierung

Übung 13: Bäume

Justin Kreikemeyer

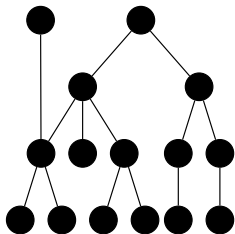
Informatik, Uni Rostock

Plan

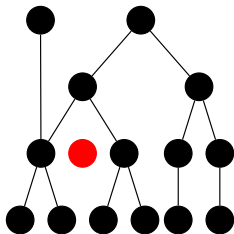
- (Binärer) Baum
- Datenstruktur: Heap
- Datenstruktur: Binärer Suchbaum

Nächste Woche: Konsultation (**bringt Fragen mit!**)

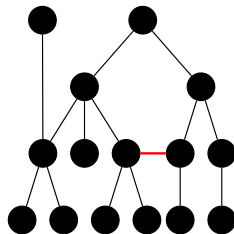
Definition: Baum



Baum!



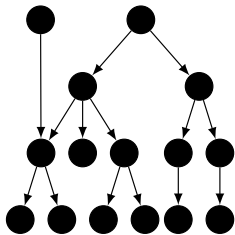
Nicht zusammenhängend!



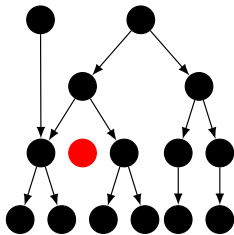
Nicht kreisfrei!

- Ein **Graph**: Knoten + Kanten, Kanten verbinden Knoten
- **Zusammenhängend**: „Jeder Knoten ist verbunden“
- **Kreisfrei**: „Es gibt keine Kreise“

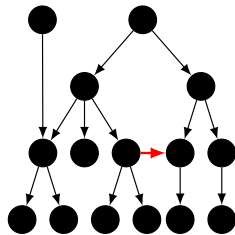
Definition: Gerichteter Baum



Gerichteter Baum!



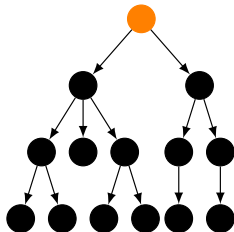
Nicht zusammenhängend!



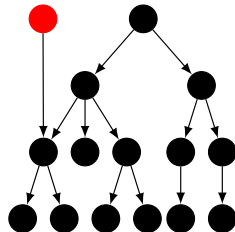
Nicht kreisfrei

- Gerichteter Baum: Die Kanten haben Richtungen!
- Einschränkung: Ohne Kantenrichtungen muss es ein Baum sein!
- (Richtung ist dabei egal)

Definition: Gewurzelter Baum



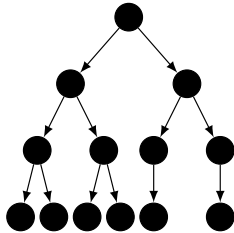
Wurzelbaum!



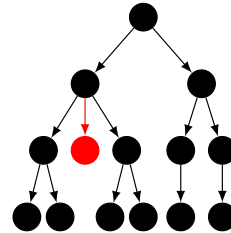
Kein Wurzelbaum!

- Wurzelbaum: Gerichteter Baum + „Startknoten“ (Wurzel)
- Einschränkung: Alle Knoten müssen von der Wurzel ausgehend erreicht werden können

Definition: Binärbaum



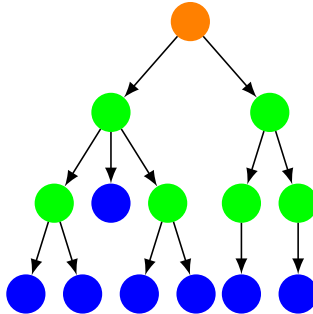
Binär!



Nicht binär!

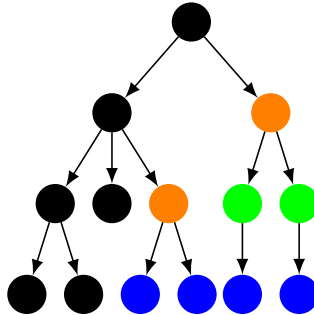
- Binärbaum = Baum + Jeder Knoten **höchstens** zwei Nachfolger

Weitere Begriffe: Wurzel + Blatt + Innerer Knoten



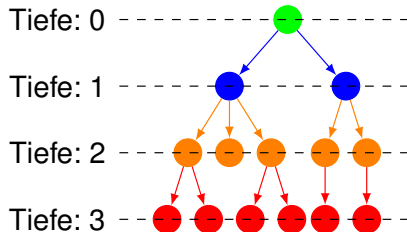
- Wurzel: Startknoten des Baumes (Orange)
- Blatt: Knoten ohne Nachfolger (Blau)
- Innerer Knoten: Weder Blatt noch Wurzel (Grün)

Weitere Begriffe: Teilbaum



- Teilbaum: Baum welcher sich aus einem ausgewählten Knoten (orange) und seinen nachfolgenden Knoten ergibt

Weitere Begriffe: Tiefe + Schichten + Höhe



- Tiefe eines Knotens: Anzahl der Schritte vom Wurzelknoten zum Knoten
- Schicht: Menge von Knoten mit der gleichen Tiefe
- Höhe:
 - Höhe des Baumes: Maximale Tiefe (Höchste Anzahl der Schritte zu einem Blatt)
 - Höhe eines Knotens: Höhe des dadurch gewählten Teilbaumes

Implementierung von (binären) Bäumen

- Idee: Ausnutzung rekursive Definition
 - Baum =
linker Baum / Knoten \ rechter Baum

Implementierung von (binären) Bäumen

```
typedef struct _node {
    int value;
    _node* left;
    _node* right;
} tree;

int main() {
    // create root of new tree
    tree* root = malloc(sizeof(tree));
    root->value = 1.0;
    root->left = NULL;
    root->right = NULL;
}
```

Wozu das Ganze? Niedrige Suchkomplexität!

Datenstruktur: (Max-)Heap

Max-Heap Eigenschaft

Ein Heap ist ein Binärbaum, bei dem alle Schichten bis auf die unterste vollständig und die Blätter der untersten Schicht linksbündig aufgefüllt sind. Zusätzlich muss der Schlüssel jedes Knotens größergleich den Schlüsseln seiner Kinderknoten sein.

- Maximum ist also immer an der Wurzel
- Verwendung z.B. für Priority-Queue
- Operationen: insert, delete, get_max
- Visualisierung (Min-Heap)

Datenstruktur: Binärer Suchbaum

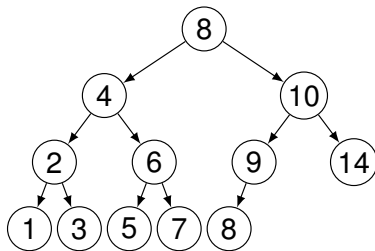
Binärer Suchbaum

Ein binärer Suchbaum ist ein binärer Baum, bei dem

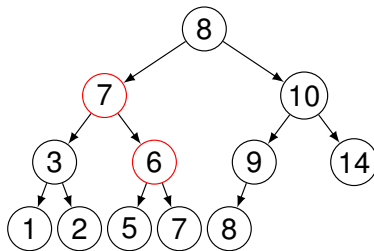
1. die Knoten des linken Teilbaums jedes Knotens nur kleinere Schlüssel als der Knoten selbst besitzen und
2. die Knoten des rechten Teilbaums jedes Knotens nur größere Schlüssel als der Knoten selbst besitzen.

- Suche benötigt also nur $O(\text{Höhe})$
- Verwendung z.B. für Tabellen
- Operationen: insert, delete, find
- **Visualisierung**

Datenstruktur: Binärer Suchbaum



Binärer Suchbaum!



Kein binärer Suchbaum!

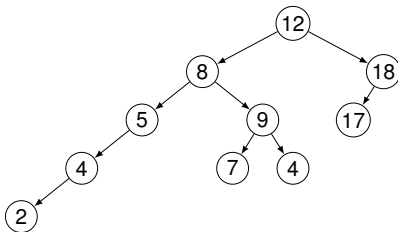
- Binärer Suchbaum = Binärbaum + Für jeden Knoten gilt
 - hat einen Wert
 - alle Werte im linken Teilbaum sind kleiner (oder gleich) dem Knotenwert
 - alle Werte im rechten Teilbaum sind größer (oder gleich) dem Knotenwert
- Anwendung: Suche - Ist ein bestimmtes Element e in meiner Folge?

Fragen?

Aufgaben

Lösen Sie die folgenden Aufgaben! Nutzen Sie dazu die Konzepte aus dieser Übung!
Weitere Aufgaben können jederzeit beim Übungsleiter erfragt werden.

Aufgaben: Baum



- Markieren Sie Wurzel, innere Knoten und Blattknoten.
- Bestimmen Sie die Höhe aller Knoten.
- Welche Tiefe hat der Baum?

Aufgaben: Heap

In der folgenden Aufgabe soll ein Heap durch die angegebenen Operationen erzeugt werden. Zeichnen Sie nach jeder Operation den jeweils resultierenden Heap und stellen Sie die Max-Heap Eigenschaft sicher:

- Hinzufügen der Knoten: (5), (2), (4), (9), (7), (5), (6), (1)
- Löschen der Wurzel

Beantworten Sie im Anschluss folgenden Aufgaben am Beispiel des resultierenden Baums:

- Bestimmen Sie die Höhen aller Knoten!
- Welche Tiefe hat der Baum?

Aufgaben: Binärer Suchbaum

In der folgenden Aufgabe soll ein binärer Suchbaum durch die angegebenen Operationen erzeugt werden. Zeichnen Sie nach jeder Operation den jeweils resultierenden binären Suchbaum:

- Hinzufügen der Knoten: (5), (2), (4), (9), (7), (5), (6), (1)
- Löschen der Wurzel

Beantworten Sie im Anschluss folgenden Aufgaben am Beispiel des resultierenden Baums:

- Bestimmen Sie die Höhen aller Knoten!
- Welche Tiefe hat der Baum?