

Informatik I: Einführung in die Programmierung

9. Bäume

Albert-Ludwigs-Universität Freiburg



UNI
FREIBURG

Peter Thiemann

4. Dezember 2019



Der Baum

Der Baum

Definition

Terminologie

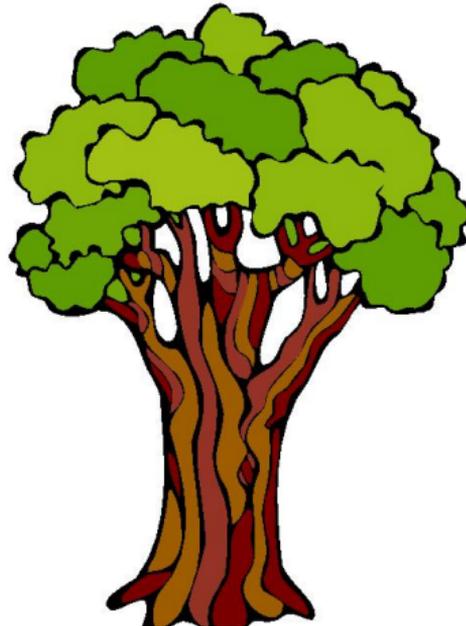
Beispiele

Binärbäume

Suchbäume



- Bäume sind in der Informatik allgegenwärtig.



Der Baum

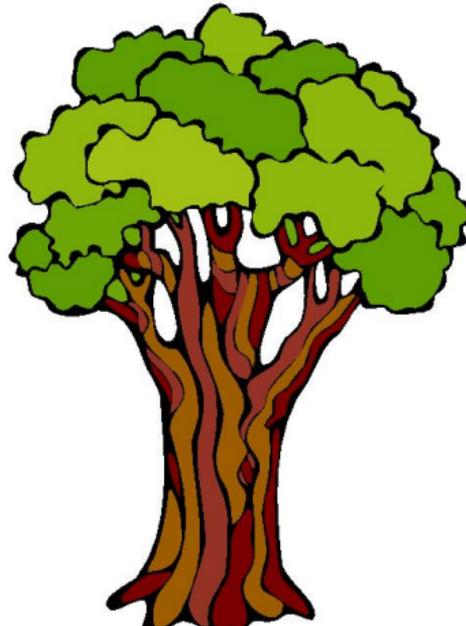
Definition
Terminologie
Beispiele

Binärbäume

Suchbäume



- Bäume sind in der Informatik allgegenwärtig.
- Gezeichnet werden sie meistens mit der Wurzel nach oben!



Der Baum

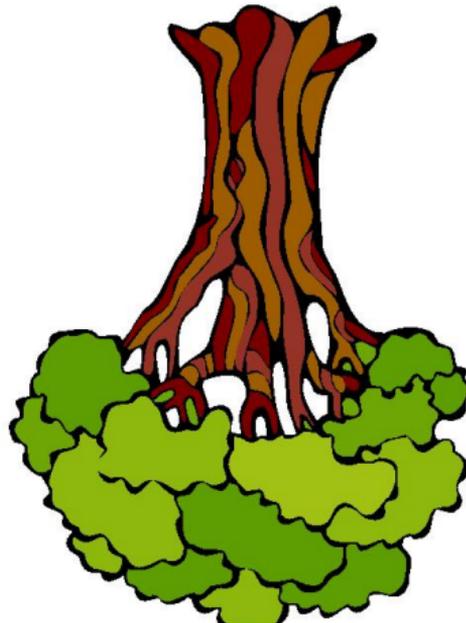
Definition
Terminologie
Beispiele

Binärbäume

Suchbäume



- Bäume sind in der Informatik allgegenwärtig.
- Gezeichnet werden sie meistens mit der Wurzel nach oben!



Der Baum

Definition

Terminologie

Beispiele

Binärbäume

Suchbäume



■ Induktive Definition:

Der Baum

Definition

Terminologie

Beispiele

Binärbäume

Suchbäume



- Induktive Definition:
 - Der **leere Baum** ist ein Baum.

Der Baum

Definition

Terminologie

Beispiele

Binärbäume

Suchbäume



■ Induktive Definition:

- Der **leere Baum** ist ein Baum.
- Wenn t_1, \dots, t_n , $n \geq 0$ disjunkte Bäume sind und k ein **Knoten**, der nicht in t_1, \dots, t_n vorkommt, dann ist auch die Struktur bestehend aus der **Wurzel** k mit **zugeordneten Teilbäumen** t_1, \dots, t_n ein **Baum**.

Der Baum

Definition

Terminologie

Beispiele

Binärbäume

Suchbäume



■ Induktive Definition:

- Der **leere Baum** ist ein Baum.
- Wenn t_1, \dots, t_n , $n \geq 0$ disjunkte Bäume sind und k ein **Knoten**, der nicht in t_1, \dots, t_n vorkommt, dann ist auch die Struktur bestehend aus der **Wurzel** k mit **zugeordneten Teilbäumen** t_1, \dots, t_n ein **Baum**.
- Nichts sonst ist ein Baum.

Der Baum

Definition

Terminologie

Beispiele

Binärbäume

Suchbäume



■ Induktive Definition:

- Der **leere Baum** ist ein Baum.
- Wenn t_1, \dots, t_n , $n \geq 0$ disjunkte Bäume sind und k ein **Knoten**, der nicht in t_1, \dots, t_n vorkommt, dann ist auch die Struktur bestehend aus der **Wurzel** k mit **zugeordneten Teilbäumen** t_1, \dots, t_n ein **Baum**.
- Nichts sonst ist ein Baum.
- Beispiel:

Der Baum

Definition

Terminologie

Beispiele

Binärbäume

Suchbäume



■ Induktive Definition:

- Der **leere Baum** ist ein Baum.
- Wenn t_1, \dots, t_n , $n \geq 0$ disjunkte Bäume sind und k ein **Knoten**, der nicht in t_1, \dots, t_n vorkommt, dann ist auch die Struktur bestehend aus der **Wurzel** k mit **zugeordneten Teilbäumen** t_1, \dots, t_n ein **Baum**.
- Nichts sonst ist ein Baum.
- Beispiel:



Der Baum

Definition

Terminologie

Beispiele

Binärbäume

Suchbäume



■ Induktive Definition:

- Der **leere Baum** ist ein Baum.
- Wenn t_1, \dots, t_n , $n \geq 0$ disjunkte Bäume sind und k ein **Knoten**, der nicht in t_1, \dots, t_n vorkommt, dann ist auch die Struktur bestehend aus der **Wurzel** k mit **zugeordneten Teilbäumen** t_1, \dots, t_n ein **Baum**.
- Nichts sonst ist ein Baum.
- Beispiel:



Der Baum

Definition

Terminologie

Beispiele

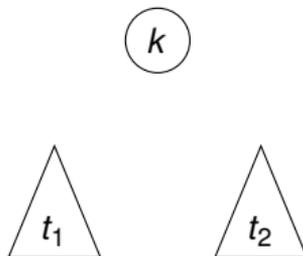
Binärbäume

Suchbäume



■ Induktive Definition:

- Der **leere Baum** ist ein Baum.
- Wenn t_1, \dots, t_n , $n \geq 0$ disjunkte Bäume sind und k ein **Knoten**, der nicht in t_1, \dots, t_n vorkommt, dann ist auch die Struktur bestehend aus der **Wurzel** k mit **zugeordneten Teilbäumen** t_1, \dots, t_n ein **Baum**.
- Nichts sonst ist ein Baum.
- Beispiel:



Der Baum

Definition

Terminologie

Beispiele

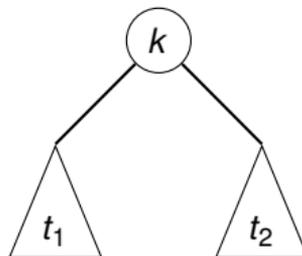
Binärbäume

Suchbäume



■ Induktive Definition:

- Der **leere Baum** ist ein Baum.
- Wenn t_1, \dots, t_n , $n \geq 0$ disjunkte Bäume sind und k ein **Knoten**, der nicht in t_1, \dots, t_n vorkommt, dann ist auch die Struktur bestehend aus der **Wurzel** k mit **zugeordneten Teilbäumen** t_1, \dots, t_n ein **Baum**.
- Nichts sonst ist ein Baum.
- Beispiel:



Der Baum

Definition

Terminologie

Beispiele

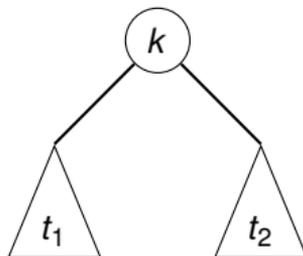
Binärbäume

Suchbäume



■ Induktive Definition:

- Der **leere Baum** ist ein Baum.
- Wenn t_1, \dots, t_n , $n \geq 0$ disjunkte Bäume sind und k ein **Knoten**, der nicht in t_1, \dots, t_n vorkommt, dann ist auch die Struktur bestehend aus der **Wurzel** k mit **zugeordneten Teilbäumen** t_1, \dots, t_n ein **Baum**.
- Nichts sonst ist ein Baum.
- Beispiel:



- **Beachte:** Bäume können auch anders definiert werden und können auch eine andere Gestalt haben (z.B. ungewurzelt).

Der Baum

Definition

Terminologie

Beispiele

Binärbäume

Suchbäume



- Alle Knoten, denen keine Teilbäume zugeordnet sind, heißen **Blätter**.

Der Baum

Definition

Terminologie

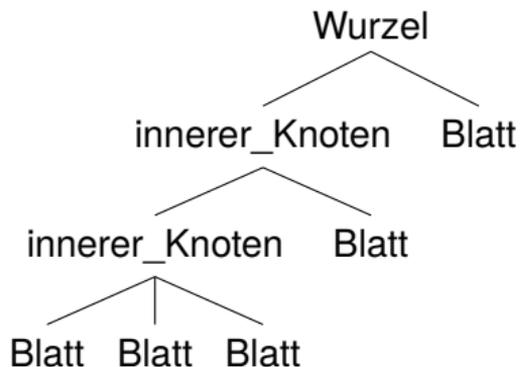
Beispiele

Binärbäume

Suchbäume



- Alle Knoten, denen keine Teilbäume zugeordnet sind, heißen **Blätter**.
- Knoten, die keine Blätter sind, heißen **innere Knoten**.



Der Baum

Definition

Terminologie

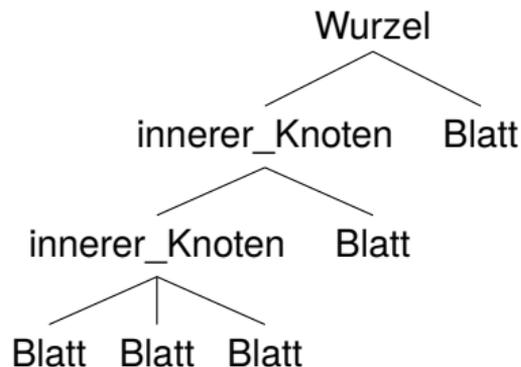
Beispiele

Binärbäume

Suchbäume



- Alle Knoten, denen keine Teilbäume zugeordnet sind, heißen **Blätter**.
- Knoten, die keine Blätter sind, heißen **innere Knoten**.



- Die Wurzel kann also ein Blatt sein (keine weiteren Teilbäume) oder ein innerer Knoten.

Der Baum

Definition

Terminologie

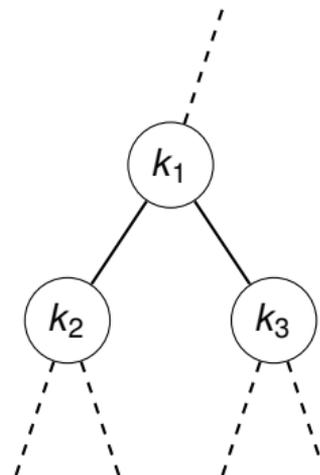
Beispiele

Binärbäume

Suchbäume



- Wenn k_1 ein Knoten und k_2 die Wurzel eines zugeordneten Teilbaums ist, dann gilt:



Der Baum

Definition

Terminologie

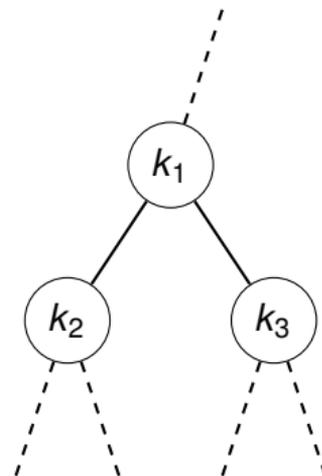
Beispiele

Binärbäume

Suchbäume



- Wenn k_1 ein Knoten und k_2 die Wurzel eines zugeordneten Teilbaums ist, dann gilt:
 - k_1 ist **Elternknoten** von k_2 ,



Der Baum

Definition

Terminologie

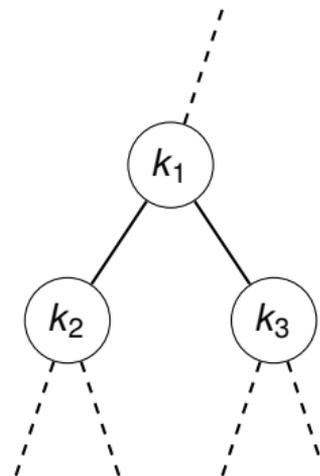
Beispiele

Binärbäume

Suchbäume



- Wenn k_1 ein Knoten und k_2 die Wurzel eines zugeordneten Teilbaums ist, dann gilt:
 - k_1 ist **Elternknoten** von k_2 ,
 - k_1 sowie der Elternknoten von k_1 sowie dessen Elternknoten usw. sind **Vorgänger** von k_2 .



Der Baum

Definition

Terminologie

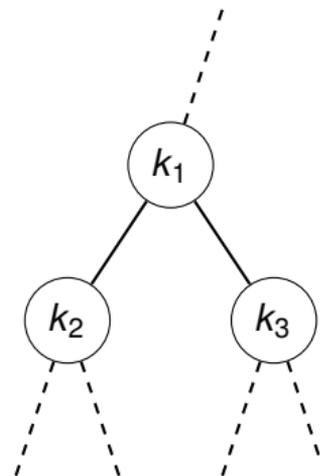
Beispiele

Binärbäume

Suchbäume



- Wenn k_1 ein Knoten und k_2 die Wurzel eines zugeordneten Teilbaums ist, dann gilt:
 - k_1 ist **Elternknoten** von k_2 ,
 - k_1 sowie der Elternknoten von k_1 sowie dessen Elternknoten usw. sind **Vorgänger** von k_2 .
 - k_2 ist **Kind** von k_1 .



Der Baum

Definition

Terminologie

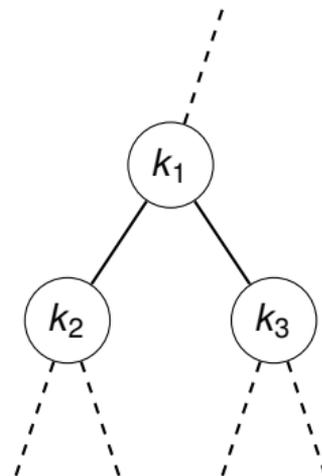
Beispiele

Binärbäume

Suchbäume



- Wenn k_1 ein Knoten und k_2 die Wurzel eines zugeordneten Teilbaums ist, dann gilt:
 - k_1 ist **Elternknoten** von k_2 ,
 - k_1 sowie der Elternknoten von k_1 sowie dessen Elternknoten usw. sind **Vorgänger** von k_2 .
 - k_2 ist **Kind** von k_1 .
 - Alle Kinder von k_1 , deren Kinder, usw. sind **Nachfolger** von k_1 .



Der Baum

Definition

Terminologie

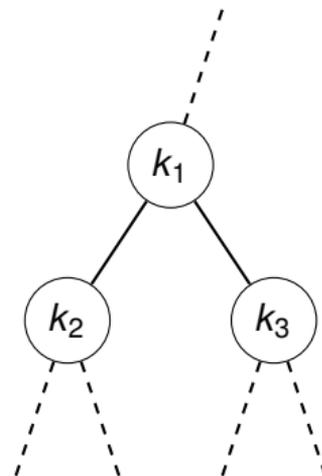
Beispiele

Binärbäume

Suchbäume



- Wenn k_1 ein Knoten und k_2 die Wurzel eines zugeordneten Teilbaums ist, dann gilt:
 - k_1 ist **Elternknoten** von k_2 ,
 - k_1 sowie der Elternknoten von k_1 sowie dessen Elternknoten usw. sind **Vorgänger** von k_2 .
 - k_2 ist **Kind** von k_1 .
 - Alle Kinder von k_1 , deren Kinder, usw. sind **Nachfolger** von k_1 .
- Bäume sind oft **markiert**. Die Markierung weist jedem Knoten eine **Marke** zu.



Der Baum

Definition

Terminologie

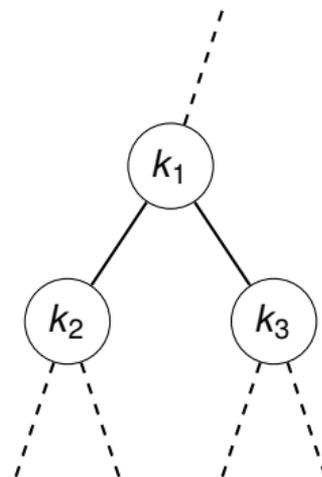
Beispiele

Binärbäume

Suchbäume



- Wenn k_1 ein Knoten und k_2 die Wurzel eines zugeordneten Teilbaums ist, dann gilt:
 - k_1 ist **Elternknoten** von k_2 ,
 - k_1 sowie der Elternknoten von k_1 sowie dessen Elternknoten usw. sind **Vorgänger** von k_2 .
 - k_2 ist **Kind** von k_1 .
 - Alle Kinder von k_1 , deren Kinder, usw. sind **Nachfolger** von k_1 .
- Bäume sind oft **markiert**. Die Markierung weist jedem Knoten eine **Marke** zu.
- Formal: Wenn K die Knotenmenge eines Baums ist und M eine Menge von Marken, dann ist die **Markierung eine Abbildung $\mu : K \rightarrow M$** .



Der Baum

Definition

Terminologie

Beispiele

Binärbäume

Suchbäume

Beispiel: Verzeichnisbaum



In Linux (und anderen Betriebssystemen) ist die Verzeichnisstruktur im Wesentlichen baumartig.

Der Baum

Definition

Terminologie

Beispiele

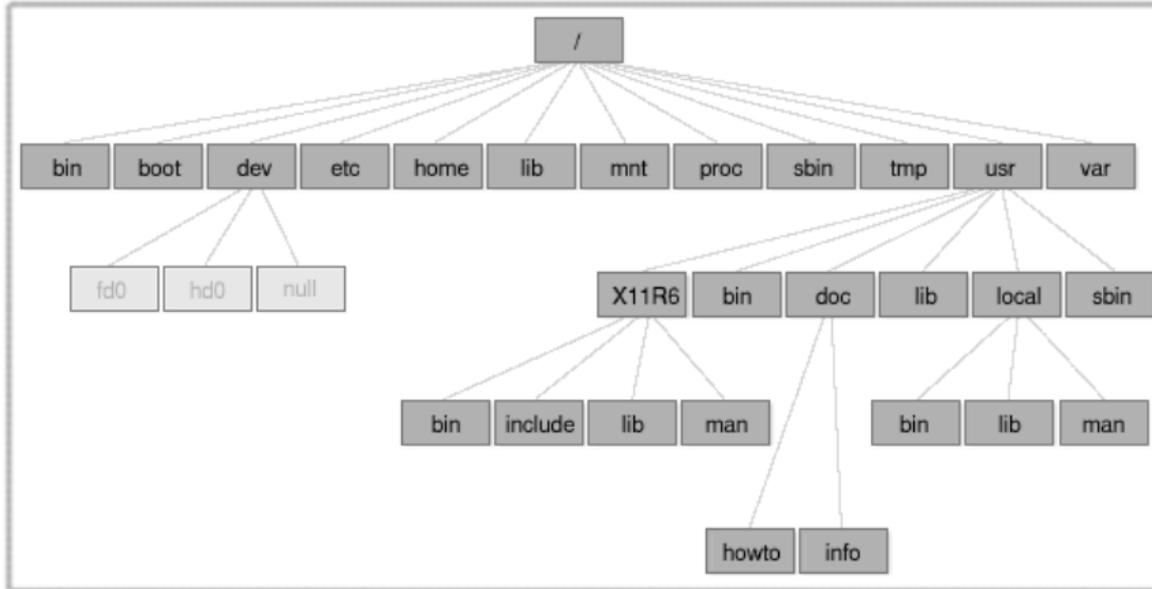
Binärbäume

Suchbäume

Beispiel: Verzeichnisbaum



In Linux (und anderen Betriebssystemen) ist die Verzeichnisstruktur im Wesentlichen baumartig.



Der Baum

Definition

Terminologie

Beispiele

Binärbäume

Suchbäume

Beispiel: Syntaxbaum



Wenn die Struktur einer Sprache mit Hilfe einer formalen Grammatiken spezifiziert ist, dann kann der Satzaufbau durch sogenannte Syntaxbäume beschrieben werden.

Der Baum

Definition

Terminologie

Beispiele

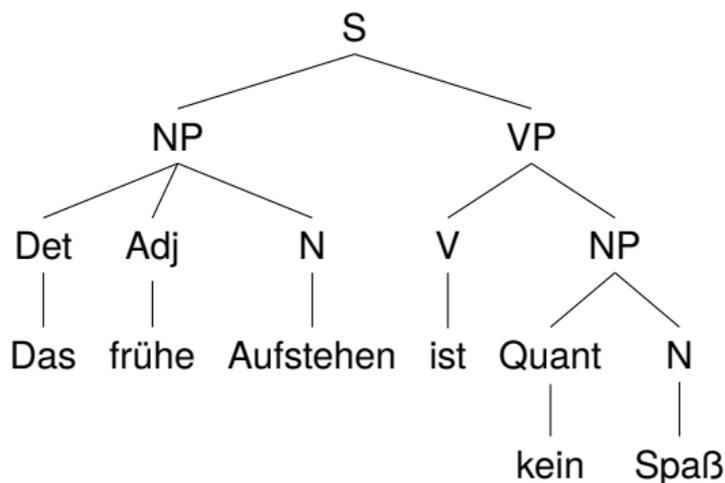
Binärbäume

Suchbäume

Beispiel: Syntaxbaum



Wenn die Struktur einer Sprache mit Hilfe einer formalen Grammatiken spezifiziert ist, dann kann der Satzaufbau durch sogenannte Syntaxbäume beschrieben werden.



Der Baum

Definition

Terminologie

Beispiele

Binärbäume

Suchbäume

Beispiel: Ausdrucksbaum



- Bäume können arithmetische (und andere) Ausdrücke so darstellen, dass ihre Auswertung eindeutig (und einfach durchführbar) ist, ohne dass Klammern notwendig sind.

Beispiel: Ausdrucksbaum



- Bäume können arithmetische (und andere) Ausdrücke so darstellen, dass ihre Auswertung eindeutig (und einfach durchführbar) ist, ohne dass Klammern notwendig sind.
- Beispiel: $(5 + 6) * 3 * 2$

Der Baum

Definition

Terminologie

Beispiele

Binärbäume

Suchbäume

Beispiel: Ausdrucksbaum

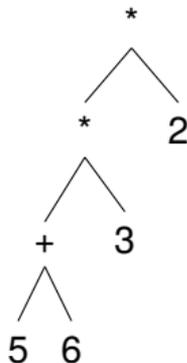


- Bäume können arithmetische (und andere) Ausdrücke so darstellen, dass ihre Auswertung eindeutig (und einfach durchführbar) ist, ohne dass Klammern notwendig sind.
- Beispiel: $(5 + 6) * 3 * 2$
- Entspricht: $((5 + 6) * 3) * 2$

Beispiel: Ausdrucksbaum

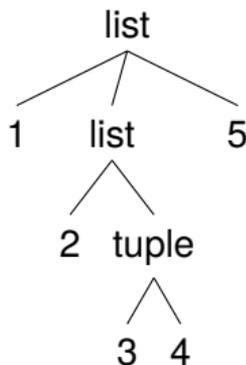


- Bäume können arithmetische (und andere) Ausdrücke so darstellen, dass ihre Auswertung eindeutig (und einfach durchführbar) ist, ohne dass Klammern notwendig sind.
- Beispiel: $(5 + 6) * 3 * 2$
- Entspricht: $((5 + 6) * 3) * 2$
- Operatoren als Markierung innerer Knoten, Zahlen als Markierung der Blätter:





- Jede Liste und jedes Tupel kann als Baum angesehen werden, bei dem der Typ die Knotenmarkierung ist und die Elemente die Teilbäume sind.
- Beispiel: $[1, [2, (3, 4)], 5]$



Der Baum

Definition

Terminologie

Beispiele

Binärbäume

Suchbäume



Binärbäume

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



- Der Binärbaum ist ein **Spezialfall eines Baumes**.

Induktive Definition

Ein **Binärbaum** ist entweder **leer** oder besteht aus einem (Wurzel-) Knoten und zwei Teilbäumen, die selbst wieder Binärbäume sind.
Nichts sonst ist ein Binärbaum.

- Im Binärbaum nennen wir einen Knoten
 - ein **Blatt**, falls beide Teilbäume leer sind
 - einen **inneren Knoten**, falls mindestens ein Teilbaum vorhanden ist
- ⇒ i.a. sind von einem Blatt keine weiteren Knoten erreichbar
- Für viele Anwendungsfälle angemessen.
 - Funktionen über solchen Bäumen sind einfach definierbar.

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



- Der **leere Baum** wird durch `None` repräsentiert.

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



- Der **leere Baum** wird durch `None` repräsentiert.
- Jeder andere **Knoten** wird durch ein `Node`-Objekt repräsentiert.

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



- Der **leere Baum** wird durch `None` repräsentiert.
- Jeder andere **Knoten** wird durch ein `Node`-Objekt repräsentiert.
- Ein `Node`-Objekt besitzt folgende Attribute

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



- Der **leere Baum** wird durch `None` repräsentiert.
- Jeder andere **Knoten** wird durch ein `Node`-Objekt repräsentiert.
- Ein `Node`-Objekt besitzt folgende Attribute
 - `mark` enthält die **Markierung**.

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



- Der **leere Baum** wird durch `None` repräsentiert.
- Jeder andere **Knoten** wird durch ein `Node`-Objekt repräsentiert.
- Ein `Node`-Objekt besitzt folgende Attribute
 - `mark` enthält die **Markierung**.
 - `left` enthält den **linken Teilbaum**.

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



- Der **leere Baum** wird durch `None` repräsentiert.
- Jeder andere **Knoten** wird durch ein `Node`-Objekt repräsentiert.
- Ein `Node`-Objekt besitzt folgende Attribute
 - `mark` enthält die **Markierung**.
 - `left` enthält den **linken Teilbaum**.
 - `right` enthält den **rechten Teilbaum**.

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



- Der **leere Baum** wird durch `None` repräsentiert.
- Jeder andere **Knoten** wird durch ein `Node`-Objekt repräsentiert.
- Ein `Node`-Objekt besitzt folgende Attribute
 - `mark` enthält die **Markierung**.
 - `left` enthält den **linken Teilbaum**.
 - `right` enthält den **rechten Teilbaum**.
- Beispiele:

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



- Der **leere Baum** wird durch `None` repräsentiert.
- Jeder andere **Knoten** wird durch ein `Node`-Objekt repräsentiert.
- Ein `Node`-Objekt besitzt folgende Attribute
 - `mark` enthält die **Markierung**.
 - `left` enthält den **linken Teilbaum**.
 - `right` enthält den **rechten Teilbaum**.
- Beispiele:
 - Der Baum bestehend aus dem einzigen Knoten mit der Markierung 8:
`Node (8, None, None)`

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



- Der **leere Baum** wird durch `None` repräsentiert.
- Jeder andere **Knoten** wird durch ein `Node`-Objekt repräsentiert.
- Ein `Node`-Objekt besitzt folgende Attribute
 - `mark` enthält die **Markierung**.
 - `left` enthält den **linken Teilbaum**.
 - `right` enthält den **rechten Teilbaum**.
- Beispiele:
 - Der Baum bestehend aus dem einzigen Knoten mit der Markierung 8:
`Node(8, None, None)`
 - Der Baum mit Wurzel '+', linkem Teilbaum mit Blatt 5, rechtem Teilbaum mit Blatt 6:
`Node('+', Node(5, None, None), Node(6, None, None))`

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



```
class Node:
    def __init__(self, mark, left, right):
        self.mark = mark
        self.left = left
        self.right = right
```

Der Baum

Binärbäume

Repräsentation

Beispiel

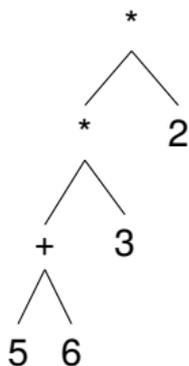
Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



wird folgendermaßen mit Node Objekten dargestellt:

```
Node ('*', Node ('*', Node ('+', Node (5, None, None),  
                             Node (6, None, None)),  
           Node (3, None, None)),  
Node (2, None, None))
```

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



Funktionsgerüst

```
def tree_str (tree : Node) -> str:
  if tree is None:
    return "fill_for_empty"
  else:
    l_str = tree_str (tree.left)
    r_str = tree_str (tree.right)
    return "fill_for_node"
```

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



Funktionsgerüst

```
def tree_str (tree : Node) -> str:
  if tree is None:
    return "fill_□for_□empty"
  else:
    l_str = tree_str (tree.left)
    r_str = tree_str (tree.right)
    return "fill_□for_□node"
```

- Node-Objekte enthalten selbst wieder Node-Objekte (oder None) in den Attributen left und right.

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



Funktionsgerüst

```
def tree_str (tree : Node) -> str:
  if tree is None:
    return "fill_for_empty"
  else:
    l_str = tree_str (tree.left)
    r_str = tree_str (tree.right)
    return "fill_for_node"
```

- Node-Objekte enthalten selbst wieder Node-Objekte (oder None) in den Attributen left und right.
- Zum Ausdrucken eines Node-Objekts müssen auch die enthaltenen Node-Objekte ausgedruckt werden.

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



Funktionsgerüst

```
def tree_str (tree : Node) -> str:
  if tree is None:
    return "fill_for_empty"
  else:
    l_str = tree_str (tree.left)
    r_str = tree_str (tree.right)
    return "fill_for_node"
```

- Node-Objekte enthalten selbst wieder Node-Objekte (oder None) in den Attributen left und right.
- Zum Ausdrucken eines Node-Objekts müssen auch die enthaltenen Node-Objekte ausgedruckt werden.
- Dafür wird tree_str **induktiv** in seiner eigenen Definition verwendet.

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



Funktionsgerüst

```
def tree_str (tree : Node) -> str:
  if tree is None:
    return "fill_for_empty"
  else:
    l_str = tree_str (tree.left)
    r_str = tree_str (tree.right)
    return "fill_for_node"
```

- Node-Objekte enthalten selbst wieder Node-Objekte (oder None) in den Attributen left und right.
- Zum Ausdrucken eines Node-Objekts müssen auch die enthaltenen Node-Objekte ausgedruckt werden.
- Dafür wird tree_str in seiner eigenen Definition verwendet.
- Ok, weil tree_str **induktiv auf den Teilbäumen** aufgerufen wird!

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



- Die **induktiven Aufrufe** `tree_str (tree.left)` und `tree_str (tree.right)` erfolgen **auf den Kindern des Knoten**.
- Ergibt sich zwangsläufig aus der induktiven Definition!
- **Induktive Aufrufe auf den Teilbäumen** sind Teil des Funktionsgerüsts, sobald eine baumartige Struktur bearbeitet werden soll.
- Die **Alternative** “`tree is None`” ergibt sich daraus, dass ein `tree` **entweder None oder ein Node-Objekt** ist.
- Alle Funktionen auf Binärbäumen verwenden dieses Gerüst.

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



```
def tree_str (tree : Node) -> str:
  if tree is None:
    return "None"
  else:
    return ("Node("
            + repr(tree.mark) + ",␣"
            + tree_str (tree.left) + ",␣"
            + tree_str (tree.right) + ")")
```

Visualisierung

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



- Die **Tiefe eines Knotens k** , $depth(k)$, ist sein Abstand von der Wurzel:

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



- Die **Tiefe eines Knotens k** , $depth(k)$, ist sein Abstand von der Wurzel:
 - 0, falls k die Wurzel ist,

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



- Die **Tiefe eines Knotens k** , $depth(k)$, ist sein Abstand von der Wurzel:
 - 0, falls k die Wurzel ist,
 - $i + 1$, wenn i die Tiefe des Elternknotens ist.

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



- Die **Tiefe eines Knotens** k , $depth(k)$, ist sein Abstand von der Wurzel:
 - 0, falls k die Wurzel ist,
 - $i + 1$, wenn i die Tiefe des Elternknotens ist.
- Die **Höhe eines Baumes** t , $height(t)$, ist die maximale Tiefe über alle Blätter:

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



- Die **Tiefe eines Knotens** k , $depth(k)$, ist sein Abstand von der Wurzel:
 - 0, falls k die Wurzel ist,
 - $i + 1$, wenn i die Tiefe des Elternknotens ist.
- Die **Höhe eines Baumes** t , $height(t)$, ist die maximale Tiefe über alle Blätter:
 - -1 für den leeren Baum,

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



- Die **Tiefe eines Knotens** k , $depth(k)$, ist sein Abstand von der Wurzel:
 - 0, falls k die Wurzel ist,
 - $i + 1$, wenn i die Tiefe des Elternknotens ist.
- Die **Höhe eines Baumes** t , $height(t)$, ist die maximale Tiefe über alle Blätter:
 - -1 für den leeren Baum,
 - $m + 1$, wenn m die maximale Höhe aller der Wurzel zugeordneten Teilbäume ist.

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



- Die **Tiefe eines Knotens** k , $depth(k)$, ist sein Abstand von der Wurzel:
 - 0, falls k die Wurzel ist,
 - $i + 1$, wenn i die Tiefe des Elternknotens ist.
- Die **Höhe eines Baumes** t , $height(t)$, ist die maximale Tiefe über alle Blätter:
 - -1 für den leeren Baum,
 - $m + 1$, wenn m die maximale Höhe aller der Wurzel zugeordneten Teilbäume ist.
- Die **Größe eines Baumes** t , $size(t)$, ist die Anzahl seiner Knoten.

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



- Die **Tiefe eines Knotens** k , $depth(k)$, ist sein Abstand von der Wurzel:
 - 0, falls k die Wurzel ist,
 - $i + 1$, wenn i die Tiefe des Elternknotens ist.
- Die **Höhe eines Baumes** t , $height(t)$, ist die maximale Tiefe über alle Blätter:
 - -1 für den leeren Baum,
 - $m + 1$, wenn m die maximale Höhe aller der Wurzel zugeordneten Teilbäume ist.
- Die **Größe eines Baumes** t , $size(t)$, ist die Anzahl seiner Knoten.
 - 0 für den leeren Baum,

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



- Die **Tiefe eines Knotens** k , $depth(k)$, ist sein Abstand von der Wurzel:
 - 0, falls k die Wurzel ist,
 - $i + 1$, wenn i die Tiefe des Elternknotens ist.
- Die **Höhe eines Baumes** t , $height(t)$, ist die maximale Tiefe über alle Blätter:
 - -1 für den leeren Baum,
 - $m + 1$, wenn m die maximale Höhe aller der Wurzel zugeordneten Teilbäume ist.
- Die **Größe eines Baumes** t , $size(t)$, ist die Anzahl seiner Knoten.
 - 0 für den leeren Baum,
 - $s + 1$, wenn s die Summe der Größen der Teilbäume ist.

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



$$\text{height}(tree) = \begin{cases} -1, & \text{if } tree \text{ is empty} \\ 1 + \max(\text{height}(tree.left), \text{height}(tree.right)), & \text{otherwise.} \end{cases}$$

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



$$\text{height}(tree) = \begin{cases} -1, & \text{if } tree \text{ is empty} \\ 1 + \max(\text{height}(tree.left), \text{height}(tree.right)), & \text{otherwise.} \end{cases}$$

$$\text{size}(tree) = \begin{cases} 0, & \text{if } tree \text{ is empty;} \\ 1 + \text{size}(tree.left) + \text{size}(tree.right), & \text{otherwise.} \end{cases}$$

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



$$\text{height}(tree) = \begin{cases} -1, & \text{if } tree \text{ is empty} \\ 1 + \max(\text{height}(tree.left), \text{height}(tree.right)), & \text{otherwise.} \end{cases}$$

$$\text{size}(tree) = \begin{cases} 0, & \text{if } tree \text{ is empty;} \\ 1 + \text{size}(tree.left) + \text{size}(tree.right), & \text{otherwise.} \end{cases}$$

Bemerkung

Die Tiefe eines Knotens kann mit dieser Baumrepräsentation nicht bestimmt werden, weil sie keinen Zugriff auf den Elternknoten erlaubt.

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



Höhe und Größe von Binärbäumen

```
def height(tree : Node) -> int:
  if (tree is None):
    return -1
  else:
    return(max(height(tree.left),
                height(tree.right)) + 1)

def size(tree : Node) -> int:
  if (tree is None):
    return 0
  else:
    return(size(tree.left)
            + size(tree.right) + 1)

tree = Node('*', Node('+', Node(6, None, None),
                        Node(5, None, None)),
            Node(1, None, None))
```

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



- Oft sollen alle Knoten eines Baumes besucht werden, wobei an jedem Knoten eine Aufgabe bearbeitet werden muss.

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



- Oft sollen alle Knoten eines Baumes besucht werden, wobei an jedem Knoten eine Aufgabe bearbeitet werden muss.
- Übliche Vorgehensweisen (**Traversierungen**) für den Besuch eines Knotens:

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



- Oft sollen alle Knoten eines Baumes besucht werden, wobei an jedem Knoten eine Aufgabe bearbeitet werden muss.
- Übliche Vorgehensweisen (**Traversierungen**) für den Besuch eines Knotens:
 - **Pre-Order** (Hauptreihenfolge): Zuerst wird der Knoten selbst bearbeitet, dann der linke, danach der rechte Teilbaum besucht

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



- Oft sollen alle Knoten eines Baumes besucht werden, wobei an jedem Knoten eine Aufgabe bearbeitet werden muss.
- Übliche Vorgehensweisen (**Traversierungen**) für den Besuch eines Knotens:
 - **Pre-Order** (Hauptreihenfolge): Zuerst wird der Knoten selbst bearbeitet, dann der linke, danach der rechte Teilbaum besucht
 - **Post-Order** (Nebenreihenfolge): Zuerst wird der linke, danach der rechte Teilbaum besucht, zum Schluss der Knoten selbst bearbeitet

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



- Oft sollen alle Knoten eines Baumes besucht werden, wobei an jedem Knoten eine Aufgabe bearbeitet werden muss.
- Übliche Vorgehensweisen (**Traversierungen**) für den Besuch eines Knotens:
 - **Pre-Order** (Hauptreihenfolge): Zuerst wird der Knoten selbst bearbeitet, dann der linke, danach der rechte Teilbaum besucht
 - **Post-Order** (Nebenreihenfolge): Zuerst wird der linke, danach der rechte Teilbaum besucht, zum Schluss der Knoten selbst bearbeitet
 - **In-Order** (symmetrische Reihenfolge): Zuerst wird der linke Teilbaum besucht, dann der Knoten bearbeitet, danach der rechte Teilbaum besucht

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



- Oft sollen alle Knoten eines Baumes besucht werden, wobei an jedem Knoten eine Aufgabe bearbeitet werden muss.
- Übliche Vorgehensweisen (**Traversierungen**) für den Besuch eines Knotens:
 - **Pre-Order** (Hauptreihenfolge): Zuerst wird der Knoten selbst bearbeitet, dann der linke, danach der rechte Teilbaum besucht
 - **Post-Order** (Nebenreihenfolge): Zuerst wird der linke, danach der rechte Teilbaum besucht, zum Schluss der Knoten selbst bearbeitet
 - **In-Order** (symmetrische Reihenfolge): Zuerst wird der linke Teilbaum besucht, dann der Knoten bearbeitet, danach der rechte Teilbaum besucht
- Manchmal auch **Reverse In-Order** (anti-symmetrische Reihenfolge): Rechter Teilbaum, Knoten, dann linker Teilbaum

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



- Oft sollen alle Knoten eines Baumes besucht werden, wobei an jedem Knoten eine Aufgabe bearbeitet werden muss.
- Übliche Vorgehensweisen (**Traversierungen**) für den Besuch eines Knotens:
 - **Pre-Order** (Hauptreihenfolge): Zuerst wird der Knoten selbst bearbeitet, dann der linke, danach der rechte Teilbaum besucht
 - **Post-Order** (Nebenreihenfolge): Zuerst wird der linke, danach der rechte Teilbaum besucht, zum Schluss der Knoten selbst bearbeitet
 - **In-Order** (symmetrische Reihenfolge): Zuerst wird der linke Teilbaum besucht, dann der Knoten bearbeitet, danach der rechte Teilbaum besucht
- Manchmal auch **Reverse In-Order** (anti-symmetrische Reihenfolge): Rechter Teilbaum, Knoten, dann linker Teilbaum
- Auch das Besuchen nach Tiefenlevel von links nach rechts (**level-order**) ist denkbar

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

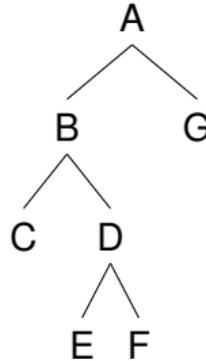
Traversierung

Zusammenfassung

Suchbäume



- Gebe Baum *pre-order* aus (Aufgabe am Knoten: Ausgabe der Markierung)



Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf

Bäumen

Baumeigenschaften

ten

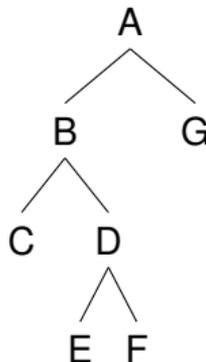
Traversierung

Zusammenfassung

Suchbäume



- Gebe Baum *pre-order* aus (Aufgabe am Knoten: Ausgabe der Markierung)



- Ausgabe: A

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf

Bäumen

Baumeigenschaften

ten

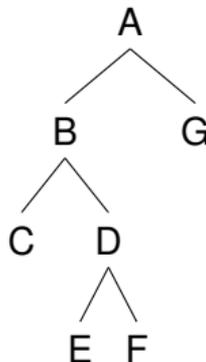
Traversierung

Zusammenfassung

Suchbäume



- Gebe Baum *pre-order* aus (Aufgabe am Knoten: Ausgabe der Markierung)



- Ausgabe: A

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf

Bäumen

Baumeigenschaften

ten

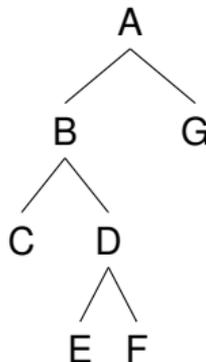
Traversierung

Zusammenfassung

Suchbäume



- Gebe Baum *pre-order* aus (Aufgabe am Knoten: Ausgabe der Markierung)



- Ausgabe: A B

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf

Bäumen

Baumeigenschaften

ten

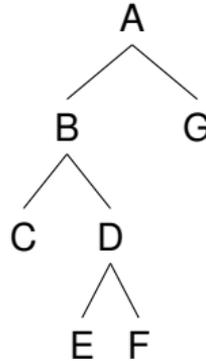
Traversierung

Zusammenfassung

Suchbäume



- Gebe Baum *pre-order* aus (Aufgabe am Knoten: Ausgabe der Markierung)



- Ausgabe: A B C

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf

Bäumen

Baumeigenschaften

ten

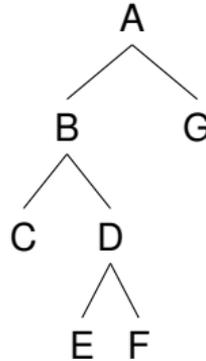
Traversierung

Zusammenfassung

Suchbäume



- Gebe Baum *pre-order* aus (Aufgabe am Knoten: Ausgabe der Markierung)



- Ausgabe: A B C D

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf

Bäumen

Baumeigenschaften

ten

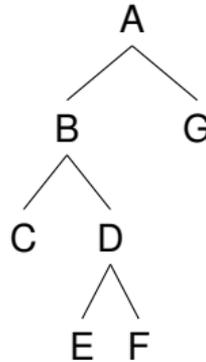
Traversierung

Zusammenfassung

Suchbäume



- Gebe Baum *pre-order* aus (Aufgabe am Knoten: Ausgabe der Markierung)



- Ausgabe: A B C D E

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf

Bäumen

Baumeigenschaften

ten

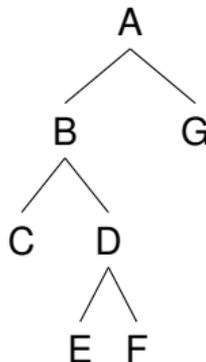
Traversierung

Zusammenfassung

Suchbäume



- Gebe Baum *pre-order* aus (Aufgabe am Knoten: Ausgabe der Markierung)



- Ausgabe: A B C D E F

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf

Bäumen

Baumeigenschaften

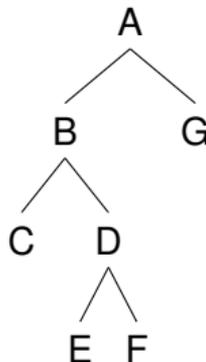
Traversierung

Zusammenfassung

Suchbäume



- Gebe Baum *pre-order* aus (Aufgabe am Knoten: Ausgabe der Markierung)



- Ausgabe: A B C D E F G

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf

Bäumen

Baumeigenschaften

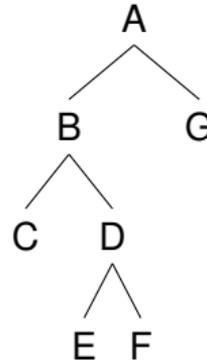
Traversierung

Zusammenfassung

Suchbäume



- Gebe Baum *post-order* aus (Aufgabe am Knoten: Ausgabe der Markierung)



Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf

Bäumen

Baumeigenschaften

ten

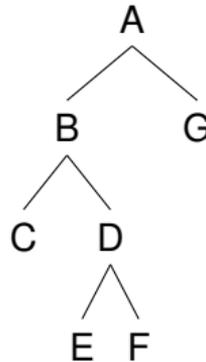
Traversierung

Zusammenfassung

Suchbäume



- Gebe Baum *post-order* aus (Aufgabe am Knoten: Ausgabe der Markierung)



- Ausgabe: C

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf

Bäumen

Baumeigenschaften

ten

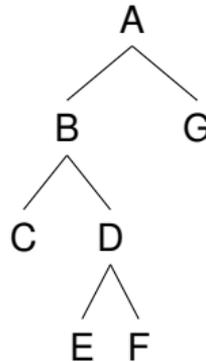
Traversierung

Zusammenfassung

Suchbäume



- Gebe Baum *post-order* aus (Aufgabe am Knoten: Ausgabe der Markierung)



- Ausgabe: C

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf

Bäumen

Baumeigenschaften

ten

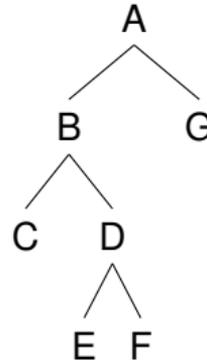
Traversierung

Zusammenfassung

Suchbäume



- Gebe Baum *post-order* aus (Aufgabe am Knoten: Ausgabe der Markierung)



- Ausgabe: C E

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf

Bäumen

Baumeigenschaften

ten

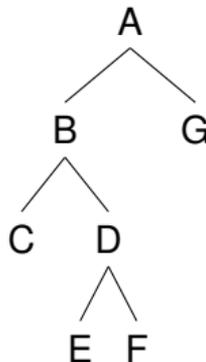
Traversierung

Zusammenfassung

Suchbäume



- Gebe Baum *post-order* aus (Aufgabe am Knoten: Ausgabe der Markierung)



- Ausgabe: C E F

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf

Bäumen

Baumeigenschaften

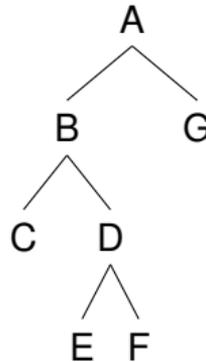
Traversierung

Zusammenfassung

Suchbäume



- Gebe Baum *post-order* aus (Aufgabe am Knoten: Ausgabe der Markierung)



- Ausgabe: C E F D

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf

Bäumen

Baumeigenschaften

ten

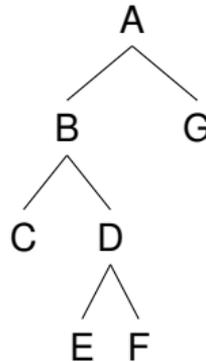
Traversierung

Zusammenfassung

Suchbäume



- Gebe Baum *post-order* aus (Aufgabe am Knoten: Ausgabe der Markierung)



- Ausgabe: C E F D B

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf

Bäumen

Baumeigenschaften

ten

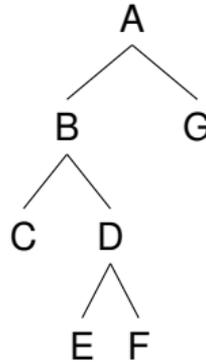
Traversierung

Zusammenfassung

Suchbäume



- Gebe Baum *post-order* aus (Aufgabe am Knoten: Ausgabe der Markierung)



- Ausgabe: C E F D B G

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf

Bäumen

Baumeigenschaften

ten

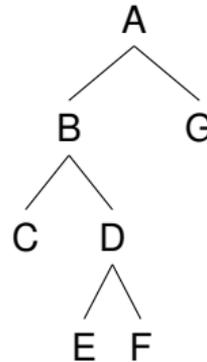
Traversierung

Zusammenfassung

Suchbäume



- Gebe Baum *post-order* aus (Aufgabe am Knoten: Ausgabe der Markierung)



- Ausgabe: C E F D B G A

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf

Bäumen

Baumeigenschaften

ten

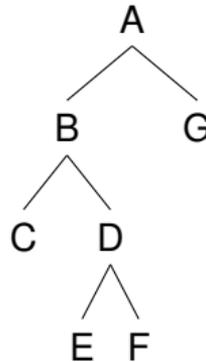
Traversierung

Zusammenfassung

Suchbäume



- Gebe Baum *in-order* aus (Aufgabe am Knoten: Ausgabe der Markierung).



Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf

Bäumen

Baumeigenschaften

ten

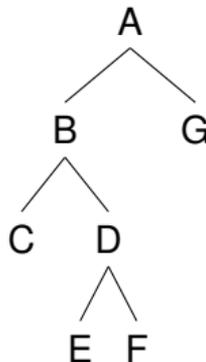
Traversierung

Zusammenfassung

Suchbäume



- Gebe Baum *in-order* aus (Aufgabe am Knoten: Ausgabe der Markierung).



- Ausgabe: C

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf

Bäumen

Baumeigenschaften

ten

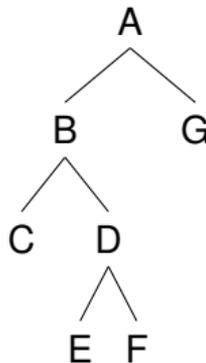
Traversierung

Zusammenfassung

Suchbäume



- Gebe Baum *in-order* aus (Aufgabe am Knoten: Ausgabe der Markierung).



- Ausgabe: C

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf

Bäumen

Baumeigenschaften

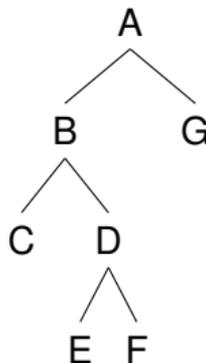
Traversierung

Zusammenfassung

Suchbäume



- Gebe Baum *in-order* aus (Aufgabe am Knoten: Ausgabe der Markierung).



- Ausgabe: C B

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf

Bäumen

Baumeigenschaften

ten

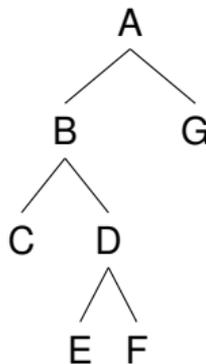
Traversierung

Zusammenfassung

Suchbäume



- Gebe Baum *in-order* aus (Aufgabe am Knoten: Ausgabe der Markierung).



- Ausgabe: C B E

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf

Bäumen

Baumeigenschaften

ten

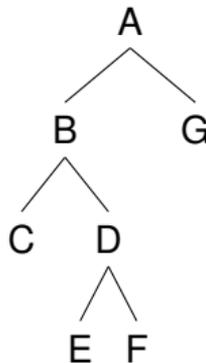
Traversierung

Zusammenfassung

Suchbäume



- Gebe Baum *in-order* aus (Aufgabe am Knoten: Ausgabe der Markierung).



- Ausgabe: C B E D

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf

Bäumen

Baumeigenschaften

ten

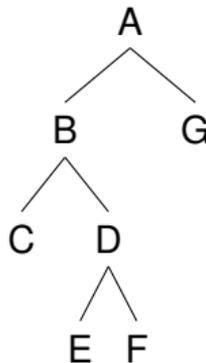
Traversierung

Zusammenfassung

Suchbäume



- Gebe Baum *in-order* aus (Aufgabe am Knoten: Ausgabe der Markierung).



- Ausgabe: C B E D F

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf

Bäumen

Baumeigenschaften

ten

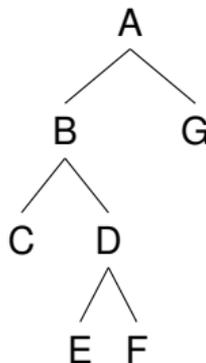
Traversierung

Zusammenfassung

Suchbäume



- Gebe Baum *in-order* aus (Aufgabe am Knoten: Ausgabe der Markierung).



- Ausgabe: C B E D F A

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf

Bäumen

Baumeigenschaften

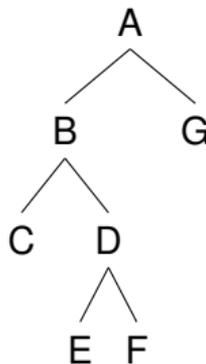
Traversierung

Zusammenfassung

Suchbäume



- Gebe Baum *in-order* aus (Aufgabe am Knoten: Ausgabe der Markierung).



- Ausgabe: C B E D F A G

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf

Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



```
def postorder(tree : Node):
    if tree is None:
        pass
    else:
        postorder(tree.left)
        postorder(tree.right)
        print(tree.mark)

def leaf (m) -> Node:
    return Node (m, None, None)

tree = Node('*', Node('+', leaf(6), leaf(5)),
              leaf(1))

postorder(tree)
```

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf

Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



```
def postorder(tree : Node):
    if tree is None:
        pass
    else:
        postorder(tree.left)
        postorder(tree.right)
        print(tree.mark)

def leaf (m) -> Node:
    return Node (m, None, None)

tree = Node('*', Node('+', leaf(6), leaf(5)),
             leaf(1))

postorder(tree)
```

Die *post-order* Ausgabe eines arithmetischen Ausdrucks heißt auch **umgekehrt polnische** oder **Postfix**-Notation (HP-Taschenrechner, Programmiersprachen *Forth* und *PostScript*)

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



- Der **Baum** ist eine Struktur, die in der Informatik allgegenwärtig ist.

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



- Der **Baum** ist eine Struktur, die in der Informatik allgegenwärtig ist.
- Verschiedene Definitionen:

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



- Der **Baum** ist eine Struktur, die in der Informatik allgegenwärtig ist.
- Verschiedene Definitionen:
 - **induktive Definition** (hier; einfach zum Programmieren),

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



- Der **Baum** ist eine Struktur, die in der Informatik allgegenwärtig ist.
- Verschiedene Definitionen:
 - **induktive Definition** (hier; einfach zum Programmieren),
 - **graphentheoretische Definition** (später; komplexere Algorithmen)

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



- Der **Baum** ist eine Struktur, die in der Informatik allgegenwärtig ist.
- Verschiedene Definitionen:
 - **induktive Definition** (hier; einfach zum Programmieren),
 - **graphentheoretische Definition** (später; komplexere Algorithmen)
- **Binärbäume** sind entweder leer oder bestehen aus einem Wurzelknoten mit genau zwei Teilbäumen, die wieder Binärbäume sind.

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



- Der **Baum** ist eine Struktur, die in der Informatik allgegenwärtig ist.
- Verschiedene Definitionen:
 - **induktive Definition** (hier; einfach zum Programmieren),
 - **graphentheoretische Definition** (später; komplexere Algorithmen)
- **Binärbäume** sind entweder leer oder bestehen aus einem Wurzelknoten mit genau zwei Teilbäumen, die wieder Binärbäume sind.
- Operationen über (Binär-)Bäumen lassen sich als **induktive Funktionen** implementieren.

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



- Der **Baum** ist eine Struktur, die in der Informatik allgegenwärtig ist.
- Verschiedene Definitionen:
 - **induktive Definition** (hier; einfach zum Programmieren),
 - **graphentheoretische Definition** (später; komplexere Algorithmen)
- **Binärbäume** sind entweder leer oder bestehen aus einem Wurzelknoten mit genau zwei Teilbäumen, die wieder Binärbäume sind.
- Operationen über (Binär-)Bäumen lassen sich als **induktive Funktionen** implementieren.
- Hauptarten der **Traversierung** von Binärbäumen: Pre-order, In-order, Post-order.

Der Baum

Binärbäume

Repräsentation

Beispiel

Funktionen auf
Bäumen

Baumeigenschaften

Traversierung

Zusammenfassung

Suchbäume



Suchbäume

Der Baum

Binärbäume

Suchbäume

Definition

Suche

Aufbau

Zusammenfassung



- Ein **Key-Value-Store** ist eine **endliche Abbildung** $W : K \hookrightarrow V$ mit
 - K eine Menge von **Schlüsseln** (*key*)
 - V eine Menge von **Werten** (*value*)
- Endlich heißt, dass die Abbildung nur für endlich viele Schlüssel $k \in K$ definiert ist (angedeutet durch \hookrightarrow für **partielle Abbildung**)
- Oft auch **Wörterbuch** (dictionary) genannt

Wörterbuch Bedeutung/Beispiel

- Ein Wörterbuch ordnet einem Wort seine Bedeutung bzw. seine Übersetzung in eine andere Sprache zu
- Schlüsselmenge = Wertemenge = Menge der Strings
- Das deutsche Wörterbuch hat nur Einträge für die (endlich vielen) Wörter der deutschen Sprache

Der Baum

Binärbäume

Suchbäume

Definition

Suche

Aufbau

Zusammenfassung



- **Einfügen** eines Schlüssels und des zugehörigen Werts
- **Suchen** nach einem Schlüssel; Rückgabe des zugehörigen Werts oder Fehlanzeige
- Weitere Operationen: **Löschen**, Werte bearbeiten, etc

Der Baum

Binärbäume

Suchbäume

Definition

Suche

Aufbau

Zusammenfassung



- **Einfügen** eines Schlüssels und des zugehörigen Werts
- **Suchen** nach einem Schlüssel; Rückgabe des zugehörigen Werts oder Fehlanzeige
- Weitere Operationen: **Löschen**, Werte bearbeiten, etc

Der Baum

Binärbäume

Suchbäume

Definition

Suche

Aufbau

Zusammenfassung

Einfachste Realisierung

- Repräsentation der Abbildung als Liste von Paaren aus Schlüssel und Wert

```
words = [("eins", "one"), ("zwei", "two"), ("drei", "tree"), ...]
```

- Einfügen: neues Paar anfügen
- Suchen: Durchlaufen der Liste; Vergleich mit erster Komponente der Paare
- Nachteil: Dauer der Suche proportional zur Anzahl der Einträge



- Suchbäume realisieren Wörterbücher und dienen dazu, Schlüssel schnell wieder zu finden.

Der Baum

Binärbäume

Suchbäume

Definition

Suche

Aufbau

Zusammenfassung



- Suchbäume realisieren Wörterbücher und dienen dazu, Schlüssel schnell wieder zu finden.
- Ein **Suchbaum** ist ein binärer Baum, bei dem **jeder Knoten k** die **Suchbaumeigenschaften** erfüllt:

Der Baum

Binärbäume

Suchbäume

Definition

Suche

Aufbau

Zusammenfassung



- Suchbäume realisieren Wörterbücher und dienen dazu, Schlüssel schnell wieder zu finden.
- Ein **Suchbaum** ist ein binärer Baum, bei dem **jeder Knoten k** die **Suchbaumeigenschaften** erfüllt:
 - Alle Markierungen im linken Teilbaum sind *kleiner* als die Markierung von k , alle Markierungen im rechten Teilbaum sind *größer*.

Der Baum

Binärbäume

Suchbäume

Definition

Suche

Aufbau

Zusammenfassung



- Suchbäume realisieren Wörterbücher und dienen dazu, Schlüssel schnell wieder zu finden.
- Ein **Suchbaum** ist ein binärer Baum, bei dem **jeder Knoten k** die **Suchbaumeigenschaften** erfüllt:
 - Alle Markierungen im linken Teilbaum sind *kleiner* als die Markierung von k , alle Markierungen im rechten Teilbaum sind *größer*.
- **Suchen nach Schlüssel m**

Der Baum

Binärbäume

Suchbäume

Definition

Suche

Aufbau

Zusammenfassung



- Suchbäume realisieren Wörterbücher und dienen dazu, Schlüssel schnell wieder zu finden.
- Ein **Suchbaum** ist ein binärer Baum, bei dem **jeder Knoten k** die **Suchbaumeigenschaften** erfüllt:
 - Alle Markierungen im linken Teilbaum sind *kleiner* als die Markierung von k , alle Markierungen im rechten Teilbaum sind *größer*.
- **Suchen nach Schlüssel m**
 - Beginne an der Wurzel

Der Baum

Binärbäume

Suchbäume

Definition

Suche

Aufbau

Zusammenfassung



- Suchbäume realisieren Wörterbücher und dienen dazu, Schlüssel schnell wieder zu finden.
- Ein **Suchbaum** ist ein binärer Baum, bei dem **jeder Knoten k** die **Suchbaumeigenschaften** erfüllt:
 - Alle Markierungen im linken Teilbaum sind *kleiner* als die Markierung von k , alle Markierungen im rechten Teilbaum sind *größer*.
- **Suchen nach Schlüssel m**
 - Beginne an der Wurzel
 - Vergleiche m mit Markierung am aktuellem Knoten,

Der Baum

Binärbäume

Suchbäume

Definition

Suche

Aufbau

Zusammenfassung



- Suchbäume realisieren Wörterbücher und dienen dazu, Schlüssel schnell wieder zu finden.
- Ein **Suchbaum** ist ein binärer Baum, bei dem **jeder Knoten k** die **Suchbaumeigenschaften** erfüllt:
 - Alle Markierungen im linken Teilbaum sind *kleiner* als die Markierung von k , alle Markierungen im rechten Teilbaum sind *größer*.
- **Suchen nach Schlüssel m**
 - Beginne an der Wurzel
 - Vergleiche m mit Markierung am aktuellem Knoten,
 - wenn gleich, stoppe und gebe True zurück,

Der Baum

Binärbäume

Suchbäume

Definition

Suche

Aufbau

Zusammenfassung



- Suchbäume realisieren Wörterbücher und dienen dazu, Schlüssel schnell wieder zu finden.
- Ein **Suchbaum** ist ein binärer Baum, bei dem **jeder Knoten k** die **Suchbaumeigenschaften** erfüllt:
 - Alle Markierungen im linken Teilbaum sind *kleiner* als die Markierung von k , alle Markierungen im rechten Teilbaum sind *größer*.
- **Suchen nach Schlüssel m**
 - Beginne an der Wurzel
 - Vergleiche m mit Markierung am aktuellem Knoten,
 - wenn gleich, stoppe und gebe True zurück,
 - wenn m kleiner ist, suche im linken Teilbaum,

Der Baum

Binärbäume

Suchbäume

Definition

Suche

Aufbau

Zusammenfassung



- Suchbäume realisieren Wörterbücher und dienen dazu, Schlüssel schnell wieder zu finden.
- Ein **Suchbaum** ist ein binärer Baum, bei dem **jeder Knoten k** die **Suchbaumeigenschaften** erfüllt:
 - Alle Markierungen im linken Teilbaum sind *kleiner* als die Markierung von k , alle Markierungen im rechten Teilbaum sind *größer*.
- **Suchen nach Schlüssel m**
 - Beginne an der Wurzel
 - Vergleiche m mit Markierung am aktuellem Knoten,
 - wenn gleich, stoppe und gebe True zurück,
 - wenn m kleiner ist, suche im linken Teilbaum,
 - wenn m größer ist, such im rechten Teilbaum.

Der Baum

Binärbäume

Suchbäume

Definition

Suche

Aufbau

Zusammenfassung



- Suchbäume realisieren Wörterbücher und dienen dazu, Schlüssel schnell wieder zu finden.
- Ein **Suchbaum** ist ein binärer Baum, bei dem **jeder Knoten k** die **Suchbaumeigenschaften** erfüllt:
 - Alle Markierungen im linken Teilbaum sind *kleiner* als die Markierung von k , alle Markierungen im rechten Teilbaum sind *größer*.
- **Suchen nach Schlüssel m**
 - Beginne an der Wurzel
 - Vergleiche m mit Markierung am aktuellem Knoten,
 - wenn gleich, stoppe und gebe True zurück,
 - wenn m kleiner ist, suche im linken Teilbaum,
 - wenn m größer ist, such im rechten Teilbaum.
- Suchzeit ist proportional zur **Höhe des Baums!**



- Suchbäume realisieren Wörterbücher und dienen dazu, Schlüssel schnell wieder zu finden.
- Ein **Suchbaum** ist ein binärer Baum, bei dem **jeder Knoten k** die **Suchbaumeigenschaften** erfüllt:
 - Alle Markierungen im linken Teilbaum sind *kleiner* als die Markierung von k , alle Markierungen im rechten Teilbaum sind *größer*.
- **Suchen nach Schlüssel m**
 - Beginne an der Wurzel
 - Vergleiche m mit Markierung am aktuellem Knoten,
 - wenn gleich, stoppe und gebe True zurück,
 - wenn m kleiner ist, suche im linken Teilbaum,
 - wenn m größer ist, such im rechten Teilbaum.
- Suchzeit ist proportional zur **Höhe des Baums!**
- Im besten Fall *logarithmisch in der Größe des Baums*.



```
def search(tree : Node, item) -> bool:
    if tree is None:
        return False
    elif tree.mark == item:
        return True
    elif tree.mark > item:
        return search(tree.left, item)
    else:
        return search(tree.right, item)

# smaller values left, bigger values in right subtree
nums = Node(10, Node(5, leaf(1), None),
            Node(15, leaf(12), leaf(20)))
print(search(nums, 12))
```

Visualisierung

Der Baum

Binärbäume

Suchbäume

Definition

Suche

Aufbau

Zusammenfassung

Aufbauen eines Suchbaums

Immutable — Vorhandene Struktur bleibt unverändert



UNI
FREIBURG

- Aufruf `insert(tree, item)` für das Einsortieren von `item` in `tree`

Der Baum

Binärbäume

Suchbäume

Definition

Suche

Aufbau

Zusammenfassung

Aufbauen eines Suchbaums

Immutable — Vorhandene Struktur bleibt unverändert



- Aufruf `insert(tree, item)` für das Einsortieren von `item` in `tree`
- Laut Definition: `tree` ist entweder leer oder ein `Node`-Objekt

Der Baum

Binärbäume

Suchbäume

Definition

Suche

Aufbau

Zusammenfassung

Aufbauen eines Suchbaums

Immutable — Vorhandene Struktur bleibt unverändert



UNI
FREIBURG

- Aufruf `insert(tree, item)` für das Einsortieren von `item` in `tree`
- Laut Definition: `tree` ist entweder leer oder ein `Node`-Objekt
- Ist `tree` leer, so wird der Knoten `leaf(item)` zurückgegeben.

Der Baum

Binärbäume

Suchbäume

Definition

Suche

Aufbau

Zusammenfassung

Aufbauen eines Suchbaums

Immutable — Vorhandene Struktur bleibt unverändert



UNI
FREIBURG

- Aufruf `insert(tree, item)` für das Einsortieren von `item` in `tree`
- Laut Definition: `tree` ist entweder leer oder ein `Node`-Objekt
- Ist `tree` leer, so wird der Knoten `leaf(item)` zurückgegeben.
- Ist `tree` ein `Node`-Objekt, so betrachte die Markierung `tree.mark`

Der Baum

Binärbäume

Suchbäume

Definition

Suche

Aufbau

Zusammenfassung

Aufbauen eines Suchbaums

Immutable — Vorhandene Struktur bleibt unverändert



Der Baum

Binärbäume

Suchbäume

Definition

Suche

Aufbau

Zusammenfassung

- Aufruf `insert(tree, item)` für das Einsortieren von `item` in `tree`
- Laut Definition: `tree` ist entweder leer oder ein `Node`-Objekt
- Ist `tree` leer, so wird der Knoten `leaf(item)` zurückgegeben.
- Ist `tree` ein `Node`-Objekt, so betrachte die Markierung `tree.mark`
 - Wenn die Marke größer als `item` ist, wird `item` in den linken Teilbaum eingesetzt und der Baum rekonstruiert (das erhält die Suchbaumeigenschaft!).

Aufbauen eines Suchbaums

Immutable — Vorhandene Struktur bleibt unverändert



Der Baum

Binärbäume

Suchbäume

Definition

Suche

Aufbau

Zusammenfassung

- Aufruf `insert(tree, item)` für das Einsortieren von `item` in `tree`
- Laut Definition: `tree` ist entweder leer oder ein `Node`-Objekt
- Ist `tree` leer, so wird der Knoten `leaf(item)` zurückgegeben.
- Ist `tree` ein `Node`-Objekt, so betrachte die Markierung `tree.mark`
 - Wenn die Marke größer als `item` ist, wird `item` in den linken Teilbaum eingesetzt und der Baum rekonstruiert (das erhält die Suchbaumeigenschaft!).
 - Falls `tree.mark` kleiner als `item` ist, entsprechend mit rechtem Teilbaum.

Aufbauen eines Suchbaums

Immutable — Vorhandene Struktur bleibt unverändert



Der Baum

Binärbäume

Suchbäume

Definition

Suche

Aufbau

Zusammenfassung

- Aufruf `insert(tree, item)` für das Einsortieren von `item` in `tree`
- Laut Definition: `tree` ist entweder leer oder ein `Node`-Objekt
- Ist `tree` leer, so wird der Knoten `leaf(item)` zurückgegeben.
- Ist `tree` ein `Node`-Objekt, so betrachte die Markierung `tree.mark`
 - Wenn die Marke größer als `item` ist, wird `item` in den linken Teilbaum eingesetzt und der Baum rekonstruiert (das erhält die Suchbaumeigenschaft!).
 - Falls `tree.mark` kleiner als `item` ist, entsprechend mit rechtem Teilbaum.
 - Falls `tree.mark == item` müssen wir nichts machen.

Aufbauen eines Suchbaums

Immutable — Vorhandene Struktur bleibt unverändert



```
def insert(tree : Node, item) -> Node:
  if tree is None:
    return leaf(item)
  elif tree.mark > item:
    return Node (tree.mark,
                 insert (tree.left, item),
                 tree.right)
  elif tree.mark < item:
    return Node (tree.mark,
                 tree.left,
                 insert(tree.right, item))
  else:
    return tree
```

Der Baum

Binärbäume

Suchbäume

Definition

Suche

Aufbau

Zusammenfassung



Der Baum

Binärbäume

Suchbäume

Definition

Suche

Aufbau

Zusammenfassung

```
def insertall (tree : Node, lst : list) -> Node:  
  for key in lst  
    tree = insert (tree, key)  
  return tree
```

```
bst = insertall (None, [10, 15, 20, 12, 5, 1])
```

Aufbauen eines Suchbaums

Mutable — Vorhandene Struktur wird verändert



```
1 def insertm(tree : Node, item) -> Node:
2   if tree is None:
3     return leaf(item)
4   if tree.mark > item:
5     tree.left = insertm(tree.left, item)
6   elif tree.mark < item:
7     tree.right = insertm(tree.right, item)
8   return tree
```

Der Baum

Binärbäume

Suchbäume

Definition

Suche

Aufbau

Zusammenfassung

Unterschied zur immutable Version

- In Zeile 5 wird der linke Teilbaum durch einen neuen Suchbaum **ersetzt**.
- In Zeile 7 wird der rechte Teilbaum durch einen neuen Suchbaum **ersetzt**.
- Diese Änderungen sind auch im Argument von `insertm` sichtbar.

Aufbauen eines Suchbaums

Mutable — Vorhandene Struktur wird verändert



Der Baum

Binärbäume

Suchbäume

Definition

Suche

Aufbau

Zusammenfassung

```
def insertmall (tree : Node, lst : list) -> Node:
  for key in lst
    tree = insertm (tree, key)
  return tree

bst = insertmall (None, [10, 15, 20, 12, 5, 1])
```



- Ein einfaches Verfahren zum Sortieren einer Liste
- Einlesen der Liste in einen Suchbaum mit `insertall`
- In-order Traversierung des Suchbaums; dabei Aufbau der Ausgabeliste von links nach rechts gemäß der Traversierung
- Die Ausgabe ist sortiert aufgrund der Suchbaumeigenschaft!

Der Baum

Binärbäume

Suchbäume

Definition

Suche

Aufbau

Zusammenfassung



- Ein **Key-Value-Store** funktioniert wie ein Wörterbuch. Er realisiert eine endliche Abbildung von Schlüsseln auf Werte.

Der Baum

Binärbäume

Suchbäume

Definition

Suche

Aufbau

Zusammenfassung



- Ein **Key-Value-Store** funktioniert wie ein Wörterbuch. Er realisiert eine endliche Abbildung von Schlüsseln auf Werte.
- Ein Key-Value-Store unterstützt mindestens die Operationen Einfügen, Suchen und Löschen.

Der Baum

Binärbäume

Suchbäume

Definition

Suche

Aufbau

Zusammenfassung



- Ein **Key-Value-Store** funktioniert wie ein Wörterbuch. Er realisiert eine endliche Abbildung von Schlüsseln auf Werte.
- Ein Key-Value-Store unterstützt mindestens die Operationen Einfügen, Suchen und Löschen.
- **Suchbäume** realisieren Key-Value-Stores.

Der Baum

Binärbäume

Suchbäume

Definition

Suche

Aufbau

Zusammenfassung



- Ein **Key-Value-Store** funktioniert wie ein Wörterbuch. Er realisiert eine endliche Abbildung von Schlüsseln auf Werte.
- Ein Key-Value-Store unterstützt mindestens die Operationen Einfügen, Suchen und Löschen.
- **Suchbäume** realisieren Key-Value-Stores.
- **Suchbäume** sind Binärbäume, die die Suchbaumeigenschaft besitzen, d.h. für jeden Knoten k befinden sich im linken Teilbaum nur kleinere, im rechten nur größere Markierungen als an k .

Der Baum

Binärbäume

Suchbäume

Definition

Suche

Aufbau

Zusammenfassung



- Ein **Key-Value-Store** funktioniert wie ein Wörterbuch. Er realisiert eine endliche Abbildung von Schlüsseln auf Werte.
- Ein Key-Value-Store unterstützt mindestens die Operationen Einfügen, Suchen und Löschen.
- **Suchbäume** realisieren Key-Value-Stores.
- **Suchbäume** sind Binärbäume, die die Suchbaumeigenschaft besitzen, d.h. für jeden Knoten k befinden sich im linken Teilbaum nur kleinere, im rechten nur größere Markierungen als an k .
- Das **Suchen** und **Einfügen** kann durch einfache rekursive Funktionen realisiert werden.

Der Baum

Binärbäume

Suchbäume

Definition

Suche

Aufbau

Zusammenfassung



- Ein **Key-Value-Store** funktioniert wie ein Wörterbuch. Er realisiert eine endliche Abbildung von Schlüsseln auf Werte.
- Ein Key-Value-Store unterstützt mindestens die Operationen Einfügen, Suchen und Löschen.
- **Suchbäume** realisieren Key-Value-Stores.
- **Suchbäume** sind Binärbäume, die die Suchbaumeigenschaft besitzen, d.h. für jeden Knoten k befinden sich im linken Teilbaum nur kleinere, im rechten nur größere Markierungen als an k .
- Das **Suchen** und **Einfügen** kann durch einfache rekursive Funktionen realisiert werden.
- **Sortierte Ausgabe** ist auch sehr einfach!

Der Baum

Binärbäume

Suchbäume

Definition

Suche

Aufbau

Zusammenfassung