

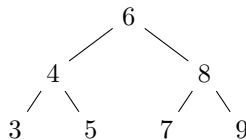
ASSIGNMENT 2 — Algorithmen und Datenstrukturen

Problem 1. Induktion und binäre Bäume

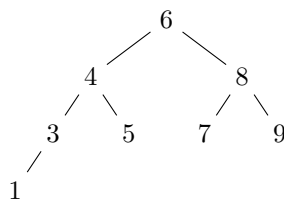
Ein binärer Baum heißt vollständig, falls jeder Knoten entweder null oder zwei Kinder besitzt.

- (a) Zeichnen Sie einen binären Suchbaum, der vollständig ist, und einen binären Suchbaum, der nicht vollständig ist.

- Balancierter vollständiger BTS:



- Unvollständiger BTS:



→ Answer

- (b) Beweisen Sie durch eine geeignete Induktion: In jedem vollständigen binären Suchbaum ist die Anzahl der Blätter genau um eins größer als die Anzahl der inneren Knoten.

- In jedem vollständigem BTS gilt: jeder Knoten n hat 0 oder genau 2 Kindknoten.
- innerer Knoten: jeder Knoten v gilt als innerer Knoten gdw. v min. 1 Kindknoten hat.
- Blätter Knoten: Jeder Knoten v gilt als Blatt, wenn v keine Kinderknoten hat.

Annahme: In jedem vollständigen binären Suchbaum ist die Anzahl der Blätter genau um eins größer als die Anzahl der inneren Knoten.

base-case BTS mit einem inneren Knoten (root):

Angenommen der BTS erfüllt die Bedingungen eines vollständigen BTS und die Werte der Knoten spielen keine Rolle, dann muss gelten:

$$\text{num}_{\text{Blätter}} = \text{num}_{\text{innerer Knoten}} + 1$$



Da die Werte in diesem Fall unerheblich sind, gilt die Zeichnung für einen vollständigen BTS mit einem inneren Knoten:

- $num_{\text{innerer Knoten}} = a$ $num_{\text{innere Knoten}} = 1$
- $num_{\text{Blätter}} = b, c$ $num_{\text{Blätter}} = 2$

Die Annahme gilt im base-case.

I.S: Wenn die Annahme bei n inneren Knoten gilt, gilt sie auch $n + 1$ inneren Knoten. Ein vollständiger BTS mit n inneren Knoten erfüllt die Gleichung:

$$num_{\text{Blätter}} = num_{\text{innerer Knoten}} + 1$$

Wenn es einen inneren Knoten zusätzlich geben soll, wird ein Blattknoten zu diesem Knoten und bekommt nach der Definition von vollständigen BTS 2 neue Kindknoten, da:

- i. nur 1 Kind Widerspruch mit Definition von vollständigen BTS.
- ii. 0 Kindknoten, dann bleibt $num_{\text{innerer Knoten}}$ gleich, da das Blatt nicht zum Inneren Knoten werden konnte.

→ **Answer**

- (c) Formulieren Sie eine ähnliche Aussage für allgemeine binäre Suchbäume und beweisen Sie sie.

Aufgabe 1 – Induktion und binäre Bäume

b) Aussage für vollständige binäre Suchbäume

In jedem vollständigen binären Suchbaum gilt: Jeder innere Knoten hat genau zwei Kindknoten.

Definitionen:

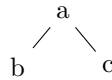
- Ein **innerer Knoten** ist ein Knoten v , der mindestens einen Kindknoten besitzt.
- Ein **Blattknoten** ist ein Knoten v , der keine Kindknoten besitzt.

Behauptung: In jedem vollständigen binären Suchbaum gilt:

$$\text{Anzahl der Blätter} = \text{Anzahl der inneren Knoten} + 1$$

Beweis durch Induktion:

Induktionsanfang (IA): Ein vollständiger binärer Baum mit genau einem inneren Knoten:



Ein innerer Knoten (a), zwei Blattknoten (b, c). Die Aussage gilt: $2 = 1 + 1$.

Induktionsvoraussetzung (IV): Die Aussage gelte für einen vollständigen Baum mit n inneren Knoten:

$$\text{Blätter} = n + 1$$

Induktionsschritt (IS): Füge einen weiteren inneren Knoten hinzu. Dies kann nur durch Umwandlung eines bisherigen Blattes erfolgen, das dann zwei neue Kinder erhält. Dadurch:

- Wird 1 Blatt zu einem inneren Knoten $\Rightarrow -1$ Blatt
- Zwei neue Blätter entstehen $\Rightarrow +2$ Blätter

Gesamtänderung: $+1$ innerer Knoten, $+1$ Blatt. Damit gilt auch für $n + 1$:

$$\text{Blätter} = (n + 1) + 1$$

c) Allgemeine Aussage für beliebige binäre Bäume

Behauptung: Für jeden Baum gilt:

$$\text{Anzahl der Knoten} = \text{Anzahl der Kanten} + 1$$

Begründung:

- Jeder Knoten (außer der Wurzel) hat genau eine eingehende Kante vom Elternknoten.
- Beim Einfügen eines neuen Knotens entsteht genau eine neue Kante.

Induktionsanfang: Ein einzelner Knoten (z.B: "a") hat 0 Kanten, 1 Knoten. Gilt: $1 = 0 + 1$.

Induktionsvoraussetzung: Für einen Baum mit n Knoten gilt:

$$n = \text{Kantenanzahl} + 1$$

Induktionsschritt: Ein neuer Knoten wird eingefügt:

- Eine neue Kante entsteht.
- Knotenanzahl wird $n + 1$

Dann gilt:

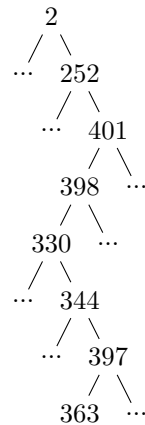
$$n + 1 = (\text{Kantenanzahl} + 1) + 1 - 1 = \text{Kantenanzahl}_{\text{neu}} + 1$$

□

Problem 2. Binäre Suchbäume

- (a) Angenommen, wir haben einen binären Suchbaum T , welcher die Zahlen von 1 bis 1000 als Schlüssel speichert. Wir suchen in T nach dem Schlüssel 363. Bestimmen Sie für jede der folgenden Schlüsselfolgen, ob sie als Folge der Einträge auf dem Suchpfad nach 363 auftreten kann. Begründen Sie jeweils Ihre Antwort.

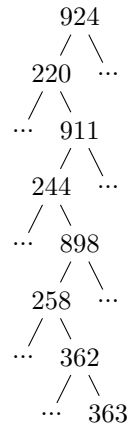
i. **2, 252, 401, 398, 330, 344, 397, 363.**



Schritt	Aktueller Knoten	Vergleich mit Ziel 363	Nächster Schritt	Intervall für nächsten Knoten
1	2	$363 > 2 \Rightarrow$ rechts	252	[3, 1000]
2	252	$363 > 252 \Rightarrow$ rechts	401	[253, 1000]
3	401	$363 < 401 \Rightarrow$ links	398	[253, 400]
4	398	$363 < 398 \Rightarrow$ links	330	[253, 397]
5	330	$363 > 330 \Rightarrow$ rechts	344	[331, 397]
6	344	$363 > 344 \Rightarrow$ rechts	397	[345, 397]
7	397	$363 < 397 \Rightarrow$ links	363	[345, 396]
8	363	Ziel erreicht	—	—

→ Answer

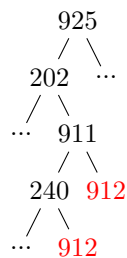
ii. **924, 220, 911, 244, 898, 258, 362, 363.**



Schritt	Aktueller Knoten	Vergleich mit Ziel 363	Nächster Schritt	Intervall für nächsten Knoten
1	924	$363 < 924 \Rightarrow$ links	220	$[1, 923]$
2	220	$363 > 220 \Rightarrow$ rechts	911	$[221, 923]$
3	911	$363 < 911 \Rightarrow$ links	244	$[221, 910]$
4	244	$363 > 244 \Rightarrow$ rechts	898	$[245, 910]$
5	898	$363 < 898 \Rightarrow$ links	258	$[245, 897]$
6	258	$363 > 258 \Rightarrow$ rechts	362	$[259, 897]$
7	362	$363 > 362 \Rightarrow$ rechts	363	$[363, 897]$
8	363	Ziel erreicht	—	—

→ Answer

iii. 925, 202, 911, 240, 912, 245, 363.

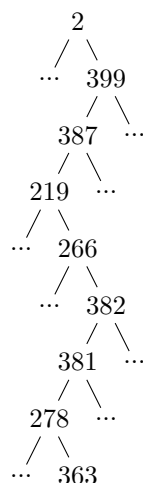


Schritt	Aktueller Knoten	Vergleich mit Ziel 363	Nächster Schritt	Intervall für nächsten Knoten
1	925	$363 < 925 \Rightarrow$ links	202	$[1, 924]$
2	202	$363 > 202 \Rightarrow$ rechts	911	$[203, 924]$
3	911	$363 < 911 \Rightarrow$ links	240	$[203, 910]$
4	240	$363 > 240 \Rightarrow$ rechts	912	$[241, 910]$
5	912	$363 < 912 \Rightarrow$ links		$912 > [910] \Rightarrow$ Fehler

- Der Knoten 912 liegt nicht mehr im Intervall von $[241, 910]$ und somit ist der Knoten nicht in der **Schlüsselfolge**.

→ Answer

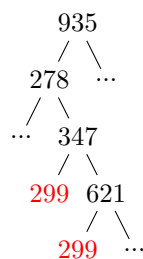
iv. 2, 399, 387, 219, 266, 382, 381, 278, 363.



Schritt	Aktueller Knoten	Vergleich mit Ziel 363	Nächster Schritt	Intervall für nächsten Knoten
1	2	$363 > 2 \Rightarrow$ rechts	399	[3, 1000]
2	399	$363 < 399 \Rightarrow$ links	387	[3, 398]
3	387	$363 < 387 \Rightarrow$ links	219	[3, 386]
4	219	$363 > 219 \Rightarrow$ rechts	266	[220, 386]
5	266	$363 > 266 \Rightarrow$ rechts	382	[267, 386]
6	382	$363 < 382 \Rightarrow$ links	381	[267, 381]
7	381	$363 < 381 \Rightarrow$ links	278	[267, 380]
8	278	$363 > 278 \Rightarrow$ rechts	363	[279, 380]
9	363	Ziel Erreicht!	-	-

→ Answer

v. 935, 278, 347, 621, 299, 392, 358, 363.



Schritt	Aktueller Knoten	Vergleich mit Ziel 363	Nächster Schritt	Intervall für nächsten Knoten
1	935	$363 < 935 \Rightarrow$ links	278	[1, 934]
2	278	$363 > 278 \Rightarrow$ rechts	347	[279, 934]
3	347	$363 > 347 \Rightarrow$ rechts	621	[348, 934]
4	621	$363 < 621 \Rightarrow$ links	299	[348, 620]
5	299	$363 > 299 \Rightarrow$ rechts		$299 < [348] \Rightarrow$ Fehler

- Der Knoten 299 liegt nicht mehr im Intervall von [348,620] und somit ist der Knoten nicht in der **Schlüsselfolge**.

→ Answer

- (b) Sei T ein binärer Baum mit n Knoten, und sei K eine total geordnete Menge von n Schlüsseln. Zeigen Sie, dass es genau eine Möglichkeit gibt, die Schlüssel aus K auf die Knoten von T zu verteilen, so dass die binäre Suchbaumeigenschaft erfüllt ist.

Problem 3. AVL-Bäume

- (a) Fügen Sie die Schlüssel A, L, G, O, D, T, S, X, Y, Z in dieser Reihenfolge in einen anfangs leeren AVL-Baum ein. Löschen Sie sodann die Schlüssel Z, A, L. Zeichnen Sie den Baum nach jedem Einfüge- und Löschvorgang, und zeigen Sie die Rotationen, welche durchgeführt werden. Annotieren Sie dabei auch die Knoten mit ihrer jeweiligen Höhe.
- (b) Beweisen Sie: Beim Einfügen in einen AVL-Baum wird höchstens eine (Einfach- oder Doppel-)Rotation ausgeführt. Gilt das auch beim Löschen (Begründung)?

Submitted by Moritz Ruge & Lennard on 09 Mai 2025.