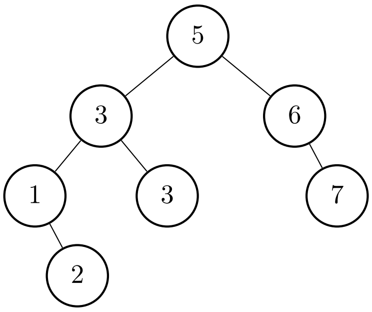
Tutoraufgabe 1 (AVL–Bäume):

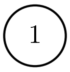
a) Fügen Sie die Schlüssel 1, 3, 5, 6, 6, 4 und 2 in dieser Reihenfolge in einen initial leeren AVL Baum ein. Geben Sie den Baum direkt nach dem Einfügen sowie nach jeder Rotation an.

b) Löschen Sie den Knoten, den die Suche nach dem Schlüssel 3 ausgibt, aus dem folgenden AVL Baum. Geben Sie den Baum direkt nach dem Löschen sowie nach jeder Rotation an.

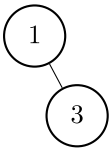


Lösung:

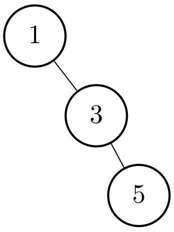
a) füge 1 ein



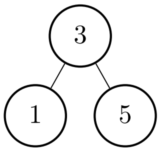
füge 3 ein



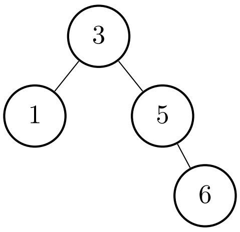
füge 5 ein



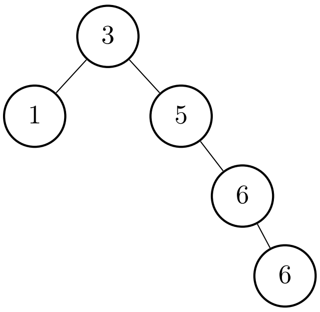
rotiere 1 nach links



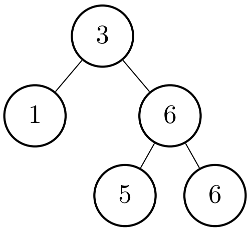
füge 6 ein



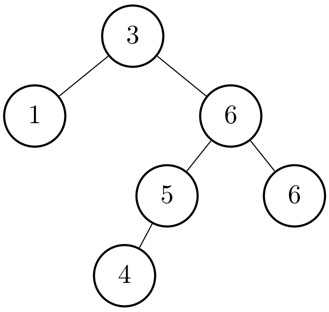
füge 6 ein



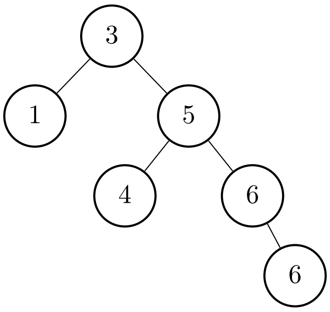
rotiere 5 nach links



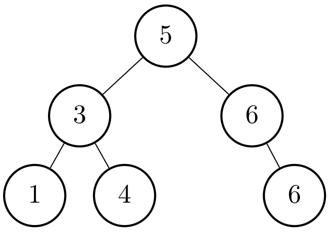
füge 4 ein



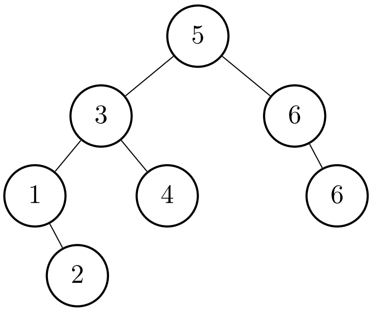
rotiere 6 nach rechts



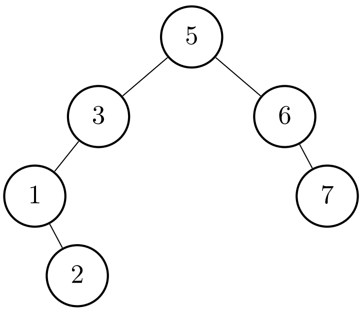
rotiere 3 nach links



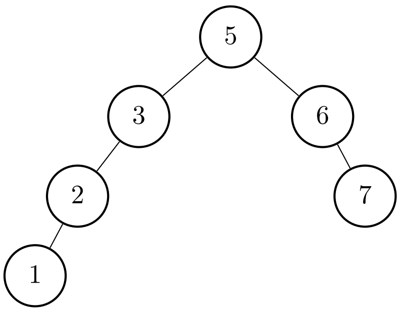
füge 2 ein



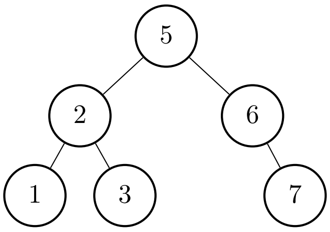
b) entferne 3



rotiere 1 nach links

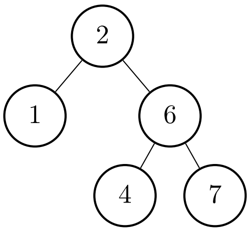


rotiere 3 nach rechts

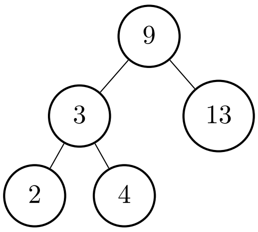


Aufgabe 2 (AVL–Bäume): (1.5 + 1.5 + 1.5 + 1.5 = 6 Punkte)

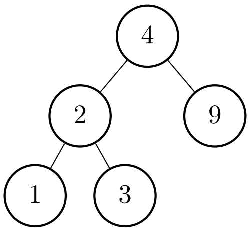
a) Fügen Sie den Schlüssel 5 in den folgenden AVL Baum ein. Geben Sie den Baum direkt nach dem Einfügen sowie nach jeder Rotation an.



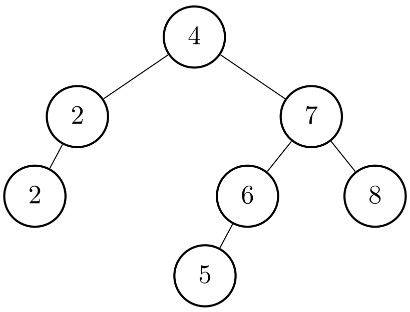
b) Fügen Sie den Schlüssel 3 in den folgenden AVL Baum ein. Geben Sie den Baum direkt nach dem Einfügen sowie nach jeder Rotation an.



c) Löschen Sie den Knoten, den die Suche nach dem Schlüssel 4 ausgibt, aus dem folgenden AVL Baum. Geben Sie den Baum direkt nach dem Löschen sowie nach jeder Rotation an.

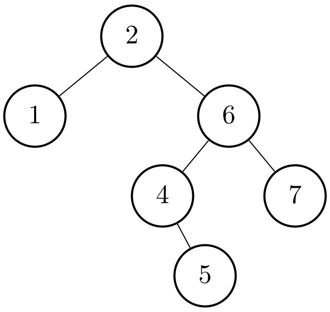


d) Löschen Sie den Knoten, den die Suche nach dem Schlüssel 2 ausgibt, aus dem folgenden AVL Baum. Geben Sie den Baum direkt nach dem Löschen sowie nach jeder Rotation an.

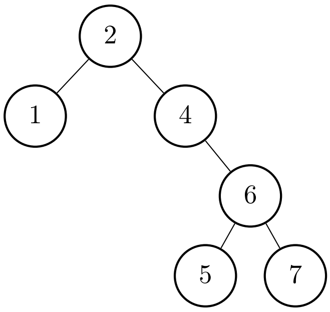


Lösung:

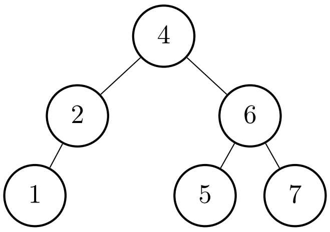
a) füge 5 ein



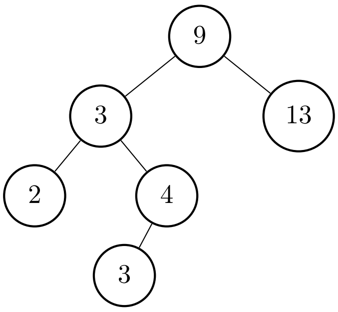
rotiere 6 nach rechts



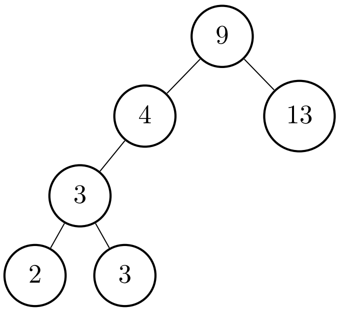
rotiere 2 nach links



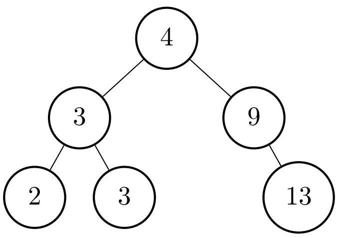
b) füge 3 ein



rotiere 3 nach links

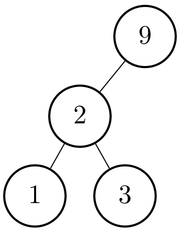


rotiere 9 nach rechts

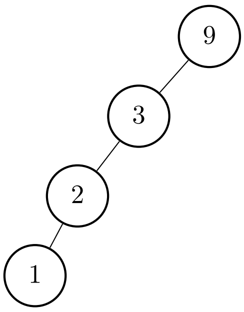


c)

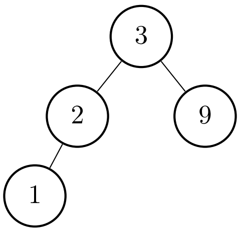
entferne 4



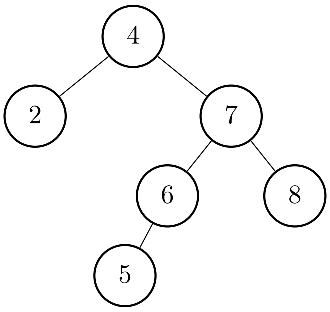
rotiere 2 nach links



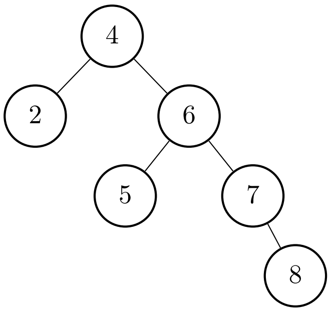
rotiere 9 nach rechts



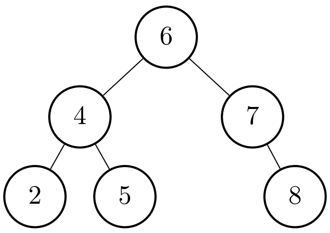
d) entferne 2



rotiere 7 nach rechts



rotiere 4 nach links



Tutoraufgabe 3 (2–3–4–Bäume):

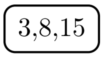
Fügen Sie in der angegebenen Reihenfolge die Schlüssel $8, 3, 15, 5, 12, 14, 10$ in einen anfangs leeren 2–3–4–Baum ein.

Lösung:

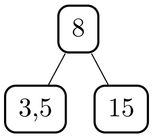
Schritt 1: Füge 8 ein

Schritt 2: Füge 3 ein

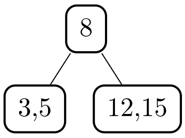
Schritt 3: Füge 15 ein



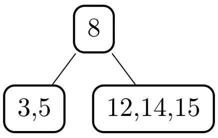
Schritt 4: Füge 5 ein



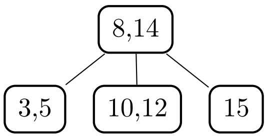
Schritt 5: Füge 12 ein



Schritt 6: Füge 14 ein

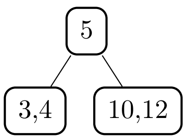


Schritt 7: Füge 10 ein



Aufgabe 4 (2–3–4–Bäume): (3 + 1 + 1 = 5 Punkte)

a) Fügen Sie in der angegebenen Reihenfolge die Schlüssel $9, 2, 14, 6, 11, 13, 0, 8, 7, 1$ in den folgenden 2–3–4–Baum ein und geben Sie die dadurch jeweils entstehenden 2–3–4–Bäume an:



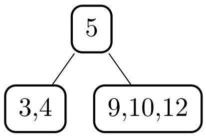
b) Geben Sie für die Zahlen 1 bis 9 eine Einfügereihenfolge in einen anfangs leeren 2–3–4–Baum an, sodass dieser seine maximale Höhe erreicht. Welche Höhe ist dies?

c) Geben Sie für die Zahlen 1 bis 9 eine Einfügereihenfolge in einen anfangs leeren 2–3–4–Baum an, sodass dieser eine minimale Anzahl an Knoten aufweist. Wie viele Knoten sind dies?

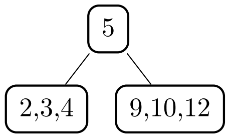
Lösung:

Schritt 1: Füge 9 ein

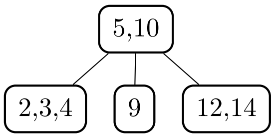
a)



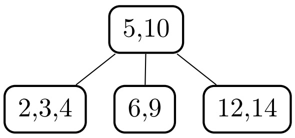
Schritt 2: Füge 2 ein



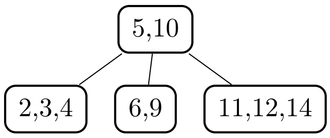
Schritt 3: Füge 14 ein



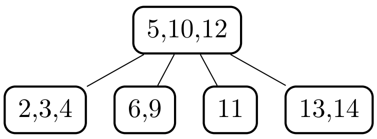
Schritt 4: Füge 6 ein



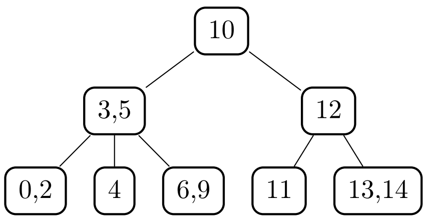
Schritt 5: Füge 11 ein



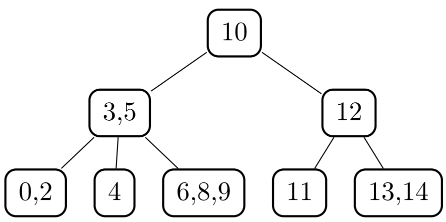
Schritt 6: Füge 13 ein



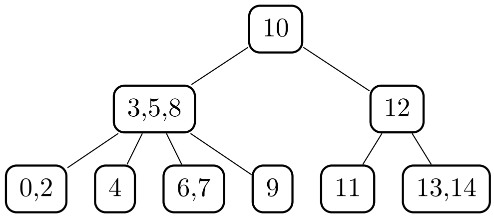
Schritt 7: Füge 0 ein



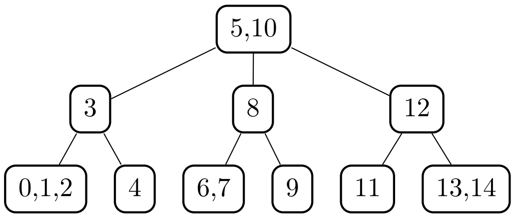
Schritt 8: Füge 8 ein



Schritt 9: Füge 7 ein



Schritt 10: Füge 1 ein



b) Die Reihenfolge $1, 2, 3, 4, 5, 6, 7, 8, 9$ führt beispielsweise zu einem 2–3–4–Baum mit Höhe 2. Ein höherer 2–3–4–Baum lässt sich mit 9 Schlüsseln nicht bilden.

c) Die Reihenfolge $4, 6, 9, 1, 2, 7, 8, 3, 5$ führt beispielsweise zu einem 2–3–4–Baum mit 4 Knoten. Ein 2–3–4–Baum mit weniger Knoten lässt sich mit 9 Schlüsseln nicht bilden.

Tutoraufgabe 5 (Knobeleien):

a) Wir haben einen binären Suchbaum, in dem die Zahlen von 1 bis 1000 einsortiert sind. Wir suchen nach der Zahl 412. Welche der folgenden Sequenzen von Schlüsselwerten kann **nicht** eine überprüfte Knotenfolge sein?

- i 998, 14, 36, 512, 247, 309, 499, 412
- ii 565, 501, 181, 673, 500, 480, 427, 412
- iii 2, 837, 547, 137, 230, 453, 402, 412
- iv 765, 699, 643, 555, 270, 315, 411, 412
- v 666, 245, 598, 301, 572, 365, 500, 412
- vi 666, 182, 575, 194, 483, 317, 595, 412
- vii 2, 781, 776, 633, 215, 545, 214, 412

b) Gegeben sei ein beliebiger binärer Suchbaum bt. Angenommen die Suche nach einem Element endet in einem Blatt r. Sei nun A die Menge der Schlüssel, die links vom Suchpfad liegen, B die Menge der Schlüssel, die auf dem Suchpfad liegen und C die Menge der Schlüssel, die rechts vom Suchpfad liegen.

Behauptung: Jedes Tripel $a \in A, b \in B, c \in C$ erfüllt $a \leq b \leq c$.

Finden Sie ein möglichst kurzes Gegenbeispiel zu der Behauptung.

Lösung:

a)
- a) 998, 14, 36, 512, 247, 309, 499, 412 $\rightarrow \checkmark$
- b) 565, 501, 181, 673, 500, 480, 427, 412 $\rightarrow \times$
- c) 2, 837, 547, 137, 230, 453, 402, 412 $\rightarrow \checkmark$

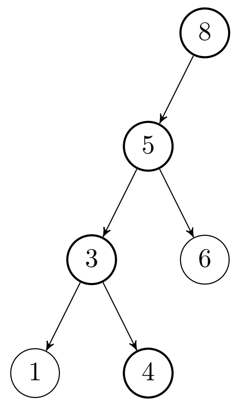
d) 765, 699, 643, 555, 270, 315, 411, 412 → ✓

e) 666, 245, 598, 301, 572, 365, 500, 412 → ✓

f) 666, 182, 575, 194, 483, 317, 595, 412 → ×

g) 2, 781, 776, 633, 215, 545, 214, 412 → ×

b) Ein Gegenbeispiel:
Suchpfad: $8 \to 5 \to 3 \to 4$

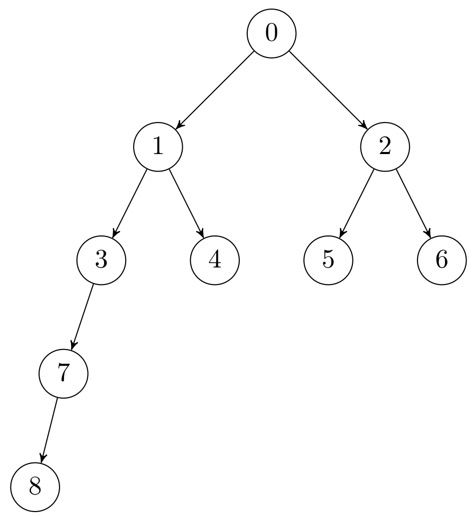


$A = \{1\}, B = \{3, 4, 5, 8\}, C = \{6\}.$

Aufgabe 6 (Knobeleien): (3 + 5 + 7* = 8 + 7* Punkte)

Ein Binärbaum ist c-höhenbalanciert, wenn der Unterschied in der Tiefe des rechten und des linken Teilbaumes jedes Knotens c nicht überschreitet.

Beispiel (2-höhenbalancierter Binärbaum):



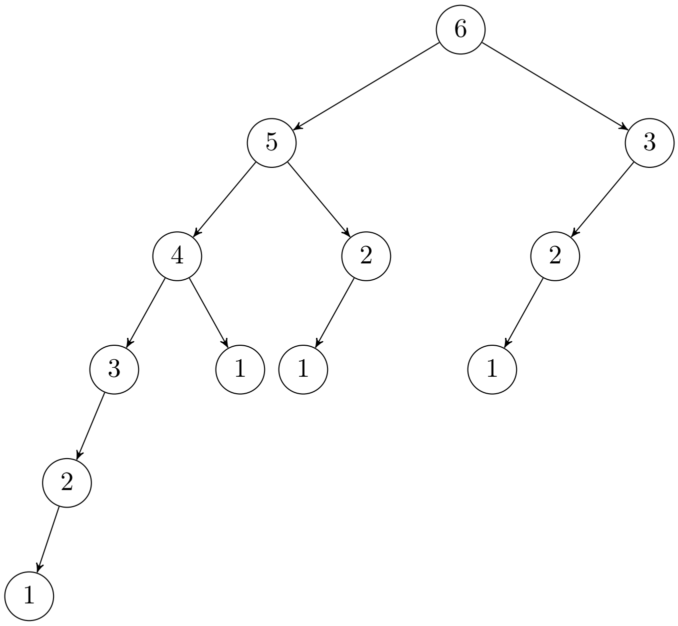
a) Geben Sie einen 2-höhenbalancierten Binärbaum der Tiefe 5 mit der minimalen Anzahl an Blättern an.

b) Beschreiben Sie eine (rekursive) Methode, um beliebige c-höhenbalancierte Binärbäume der Tiefe k mit minimaler Anzahl an Blättern zu erstellen.

c)* Zeigen Sie, dass Ihre Methode c-höhenbalancierte Binärbäume mit der minimalen Anzahl an Blättern erzeugt.

Lösung:

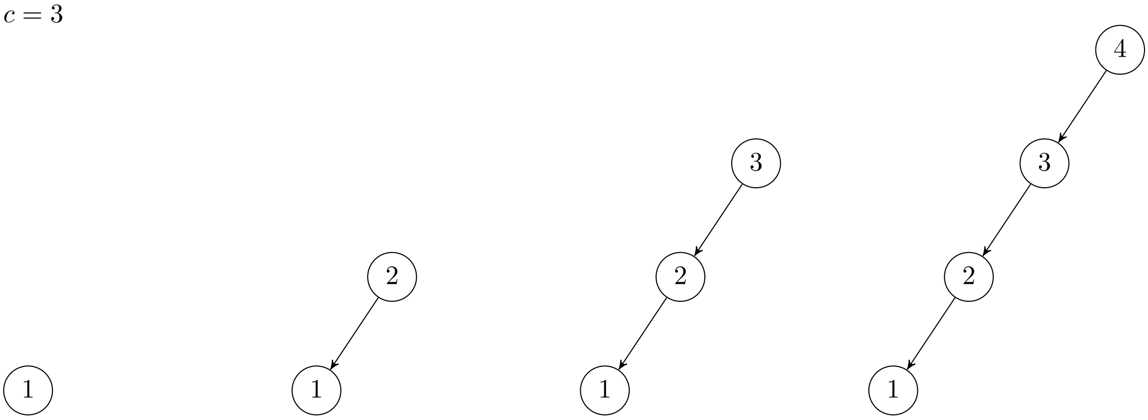
a) Ein 2-höhenbalancierter Binärbaum der Höhe 5 mit minimaler Anzahl von Blättern kann wie folgt aussehen:



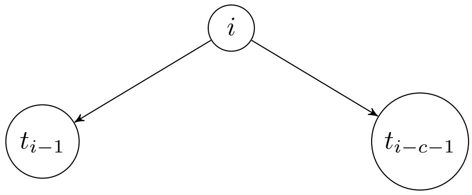
b) Eine Vorschrift für die Konstruktion von c-höhenbalancierten Binärbäumen für eine gegebene Baumhöhe k orientiert sich an der Methode für die Konstruktion von Fibonnacci–Bäumen (welche 1-höhenbalancierte Binärbäume mit minimaler Anzahl an Blättern sind):

Die Teilbäume t_i der Höhe $i \leq c$ können als lineare Strukturen erstellt werden, sie bilden die Menge an Teilbäumen, welche die Bedingung der c-Höhenbalanciertheit nicht verletzen und haben als lineare Strukturen mit nur einem Blatt die minimale Anzahl an Blättern.

Beispiel:
$c = 3$



Alle weiteren Teilbäume $t_i, i > c$ werden wie folgt aufgebaut:



c)* Es ist zu zeigen: Die Anzahl der Blätter b_i des Teilbaums t_i, $b_i = b(t_i)$ ist monoton steigend bei steigender Baumhöhe i.

Die Fälle für $i \leq c$ sind trivial (die Anzahl der Blätter ist immer 1). Im Folgenden gelte $i > c$.

Behauptung: $b(t_i) \leq b(t_{i+1}), i > c, i \in \mathbb{N}$.

Beweis:
Induktionsanfang: Laut Konstruktion gilt: $b(t_a) = 1, 0 \leq a \leq c$ und damit $b(t_{c+1}) = b(t_c) + b(t_0) = 2 \geq b(t_c)$
Induktionsschritt: Laut Konstruktion gilt: $b(t_{i+1}) = b(t_i) + b(t_{i-c}), i > c$. Mit Induktionsanfang gilt: $b(t_i) > 1, i > c$. Daraus folgt: $b(t_i) + b(t_{i-c}) > b(t_i)$. □

Durch Induktion und unter Zuhilfenahme der Tatsache, dass die Anzahl der Blätter monoton steigend ist bei steigender Baumhöhe lässt sich zeigen, dass die Konstruktion aus b) einen c–höhenbalancierten Binärbaum mit minimaler Anzahl an Blättern für eine gegebene Höhe k erzeugt:
Behauptung: $b(t_{i+1}) = b(t_i) + b(t_{i-c})$ minimal für $i + 1$.

Beweis:
Induktionsanfang: $i = c+1 \rightarrow b(t_{c+1}) = b(t_c)+b(t_0) = 2$. Wir können kein Blatt in dem resultierenden Baum entfernen, ohne nicht eine der folgenden Bedingungen zu falsifizieren:

- Der Baum hat die Höhe $c + 1$.
- Der Baum ist c-höhenbalanciert.

Induktionsschritt: Der zu konstruierende Baum soll die Höhe $i + 1$ haben. Wir kennen den Teilbaum mit Höhe i, welcher die minimale Anzahl an Blättern hat (t_i) und nutzen diesen als linken (oder rechten) Teilbaum des Baumes der Höhe $i + 1$, indem wir t_i an die Wurzel hängen. Dadurch ist die c-Höhenbalanciertheit verletzt, da der Unterschied zwischen den beiden Teilbäumen der Wurzel i ist ($i > c$). Der kleinste Teilbaum $t_{i'}$, welcher die Bedingung herstellt ist $t_{i'} = t_{i-c-1}$, da die Blattzahl monoton steigend ist. □