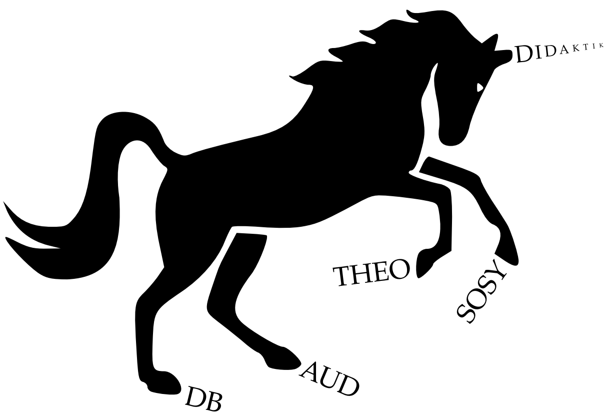
66115
Herbst
2012

Theoretische Informatik / Algorithmen (vertieft)

Aufgabenstellungen mit Lösungsvorschlägen



Die Bschlangaul-Sammlung

Hermine Bschlangaul and Friends

Aufgabenübersicht

Die Bschlangaul-Sammlung
Hermine Bschlangaul and Friends

Eine freie Aufgabensammlung mit Lösungen von Studierenden für Studierende zur Vorbereitung auf die 1. Staatsexamensprüfungen des Lehramts Informatik in Bayern.

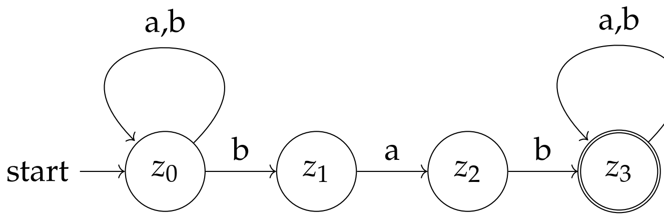
Thema Nr. 1

Aufgabe 1 [NEA und Minimalisierung]

Wir fixieren das Alphabet $\Sigma = \{a, b\}$ und definieren $L \subseteq \Sigma^*$ durch

$$L = \{\, w \mid \text{in } w \text{ kommt das Teilwort } \texttt{bab} \text{ vor} \,\}$$

z. B. ist `babaabb` $\in L$, aber `baabaabb` $\notin L$. Der folgende nichtdeterministische Automat A erkennt L:

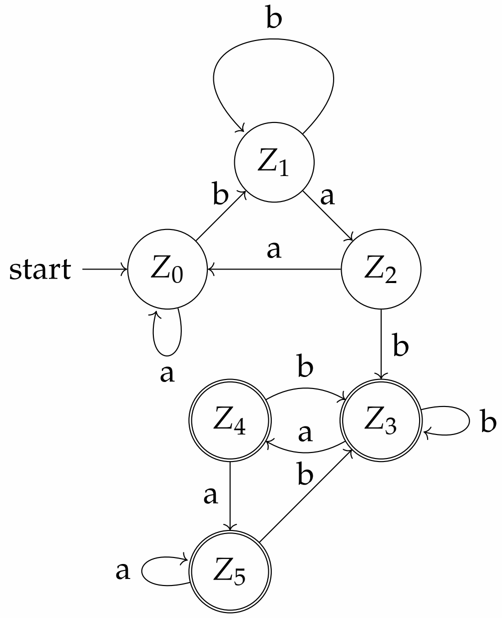


Der Automat auf flaci.com (FLACI: Formale Sprachen, abstrakte Automaten, Compiler und Interpreter) Ein Projekt der Hochschule Zittau/Görlitz und der Pädagogischen Hochschule Schwyz: flaci.com/Af75jwj3r

(a) Wenden Sie die Potenzmengenkonstruktion auf den Automaten an und geben Sie den resultierenden deterministischen Automaten an. Nicht erreichbare Zustände sollen nicht dargestellt werden.

Lösungsvorschlag

Zustandsmenge	Eingabe a	Eingabe b
Z_0 $\{z_0\}$	Z_0 $\{z_0\}$	Z_1 $\{z_0, z_1\}$
Z_1 $\{z_0, z_1\}$	Z_2 $\{z_0, z_2\}$	Z_1 $\{z_0, z_1\}$
Z_2 $\{z_0, z_2\}$	Z_0 $\{z_0\}$	Z_3 $\{z_0, z_1, z_3\}$
Z_3 $\{z_0, z_1, z_3\}$	Z_4 $\{z_0, z_2, z_3\}$	Z_3 $\{z_0, z_1, z_3\}$
Z_4 $\{z_0, z_2, z_3\}$	Z_5 $\{z_0, z_3\}$	Z_3 $\{z_0, z_1, z_3\}$
Z_5 $\{z_0, z_3\}$	Z_5 $\{z_0, z_3\}$	Z_3 $\{z_0, z_1, z_3\}$



Der Automat auf flaci.com (FLACI: Formale Sprachen, abstrakte Automaten, Compiler und Interpreter) Ein Projekt der Hochschule Zittau/Görlitz und der Pädagogischen Hochschule Schwyz: flaci.com/Aro483e89

(b) Konstruieren Sie aus dem so erhaltenen deterministischen Automaten den Minimalautomaten für L. Beschreiben Sie dabei die Arbeitsschritte des verwendeten Algorithmus in nachvollziehbarer Weise.

Lösungsvorschlag

z_0	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
z_1	x_3	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
z_2	x_2	x_2	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
z_3	x_1	x_1	x_1	$\emptyset$	$\emptyset$	$\emptyset$
z_4	x_1	x_1	x_1		$\emptyset$	$\emptyset$
z_5	x_1	x_1	x_1			$\emptyset$
	z_0	z_1	z_2	z_3	z_4	z_5

x_1 Paar aus End-/ Nicht-Endzustand kann nicht äquivalent sein.

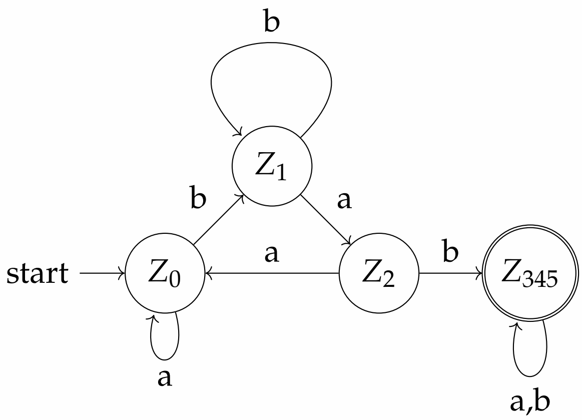
x_2 Test, ob man mit der Eingabe zu einem bereits markiertem Paar kommt.

x_3 In weiteren Iterationen markierte Zustände.

x_4 ...

Übergangstabelle

Zustandspaar	a	b
(z_0, z_1)	$(z_0, z_2)\ x_3$	(z_1, z_1)
(z_0, z_2)	(z_0, z_0)	$(z_1, z_3)\ x_2$
(z_1, z_2)	$(z_2, z_0)\ x_3$	$(z_1, z_3)\ x_2$
(z_3, z_4)	(z_4, z_5)	(z_3, z_3)
(z_3, z_5)	(z_4, z_5)	(z_3, z_3)
(z_4, z_5)	(z_5, z_5)	(z_3, z_3)



Der Automat auf flaci.com (FLACI: Formale Sprachen, abstrakte Automaten, Compiler und Interpreter) Ein Projekt der Hochschule Zittau/Görlitz und der Pädagogischen Hochschule Schwyz: flaci.com/Ar3joif5z

Aufgabe 4 [maximale Teilsumme]

Gegeben ist ein Array a von ganzen Zahlen der Länge n, z. B. :

i	0	1	2	3	4	5	6	7	8	9
a_i	5	-6	4	2	-5	7	-2	-7	3	5

Im Beispiel ist also $n = 10$. Es soll die maximale Teilsumme berechnet werden, also der Wert des Ausdrucks

$$\max_{i,j \leq n} \sum_{k=1}^{j-1} a_k$$

Im Beispiel ist dieser Wert 8 und wird für $i = 8, j = 10$ erreicht. Entwerfen Sie ein Divide-And-Conquer Verfahren, welches diese Aufgabenstellung in Zeit $\mathcal{O}(n \log n)$ löst. Skizzieren Sie Ihre Lösung hinreichend detailliert.

Tipp: Sie sollten ein geringfügig allgemeineres Problem lösen, welches neben der maximalen Teilsumme auch noch die beiden „maximalen Randsummen" berechnet. Die werden dann bei der Endausgabe verworfen.

```java
/**
 * Klasse zur Berechnung der maximalen Teilsumme einer Zahlenfolge.
 *
 * nach Teilsumme.java Klasse mit Algorithmen für die Berechnung des größten
 * gemeinsamen Teilers zweier Ganzzahlen Algorithmen und Datenstrukturen,
 * Auflage 4, Kapitel 2.1
 *
 * nach Prof. Grude, Prof. Solymosi, (c) 2000-2008: 22. April 2008
 * <a href="http://public.beuth-hochschule.de/oo-
↪  plug/A&D/prog/kap21/Teilsumme.java">Teilsumme.java</a>
 */
public class Teilsumme {

  /**
   * Berechne die maximale Teilsumme an der rechten Grenze. Die Eingabeparameter
   * müssen diese Werte aufweisen: 0 &lt;= links &lt;= rechts &lt; folge.length.
   *
   * @param folge  Die Zahlenfolge, in der die maximale Zahlensumme gerechnet
   *               werden soll.
   * @param links  Die Index-Nummer der linken Grenze.
   * @param rechts Die Index-Nummer der rechten Grenze.
   *
   * @return Die maximale Teilsumme.
   */
  private static int berechneRandRechts(int[] folge, int links, int rechts) {
    int max = 0;
    int sum = 0;
    for (int i = rechts; i >= links; i--) {
      sum += folge[i];
      max = Math.max(max, sum);
    }
    return max;
  }

  /**
   * Berechne die maximale Teilsumme an der linken Grenze. Die Eingabeparameter
   * müssen diese Werte aufweisen: 0 &lt;= links &lt;= rechts &lt; folge.length.
   *
   * @param folge  Die Zahlenfolge, in der die maximale Zahlensumme gerechnet
   *               werden soll.
   * @param links  Die Index-Nummer der linken Grenze.
   * @param rechts Die Index-Nummer der rechten Grenze.
   *
   * @return Die maximale Teilsumme.
   */
  private static int berechneRandLinks(int[] folge, int links, int rechts) {
    int max = 0;
    int sum = 0;
    for (int i = links; i <= rechts; i++) {
      sum += folge[i];
      max = Math.max(max, sum);
    }
    return max;
  }
```

```java
  /**
   * Berechne die maximale Teilsumme in der Zahlenfolge zwischen einer gegeben
   * linken und rechten Grenze. Die Eingabeparameter müssen diese Werte aufweisen:
   * 0 &lt;= links &lt;= rechts &lt; folge.length.
   *
   * @param folge  Die Zahlenfolge, in der die maximale Zahlensumme gerechnet
   *               werden soll.
   * @param links  Die Index-Nummer der linken Grenze.
   * @param rechts Die Index-Nummer der rechten Grenze.
   *
   * @return Die maximale Teilsumme.
   */
  private static int berechne(int[] folge, int links, int rechts) {
    if (links == rechts) // nur ein Element
      return Math.max(0, folge[links]);
    else {
      final int mitte = (rechts + links) / 2;
      final int maxLinks = berechne(folge, links, mitte);
      final int maxRechts = berechne(folge, mitte + 1, rechts);
      final int maxGrenzeRechts = berechneRandRechts(folge, links, mitte);
      // linke Hälfte
      final int maxGrenzeLinks = berechneRandLinks(folge, mitte + 1, rechts);
      // rechte Hälfte
      return Math.max(maxRechts, Math.max(maxLinks, maxGrenzeRechts + maxGrenzeLinks));
    }
  }

  /**
   * Berechne die maximale Teilsumme einer Zahlenfolge rekursiv mit
   * logarithmischer Zeitkomplexität.
   *
   * @param folge Die Zahlenfolge, in der die maximale Zahlensumme gerechnet
   *              werden soll.
   *
   * @return Die maximale Teilsumme.
   */
  public static int berechne(int[] folge) {
    return berechne(folge, 0, folge.length - 1);
  }

  public static void main(String[] args) {
    int[] folge = { 5, -6, 4, 2, -5, 7, -2, -7, 3, 5 };
    int ergebnis = berechne(folge);
    System.out.println(ergebnis);
  }
}
```

Code-Beispiel auf Github ansehen: `src/main/java/org/bschlangaul/examen/examen_66115/jahr_2012/herbst/Teilsumme.java`

Thema Nr. 2

Aufgabe 4: Komplexität [limes]

Gegeben seien die Funktionen $f\colon \mathbb{N} \to \mathbb{N}$ und $g\colon \mathbb{N} \to \mathbb{N}$, wobei $f(n) = (n-1)^3$ und $g(n) = (2n+3)(3n+2)$. Geben Sie an, welche der folgenden Aussagen gelten. Beweisen Sie Ihre Angaben.

(a) $f(n) \in \mathcal{O}(g(n))$

(b) $g(n) \in \mathcal{O}(f(n))$

> **Exkurs: Regel von L'Hospital**
> Die Regel von de L'Hospital ist ein Hilfsmittel zum Berechnen von Grenzwerten bei Brüchen $\frac{f}{g}$ von Funktionen f und g, wenn Zähler und Nenner entweder beide gegen 0 oder beide gegen (+ oder -) unendlich gehen. Wenn in einem solchen Fall auch der Grenzwert des Bruches der Ableitungen existiert, so hat dieser denselben Wert wie der ursprüngliche Grenzwert: [a]
>
> $$\lim_{x \to x_0} \frac{f(x)}{g(x)} = \lim_{x \to x_0} \frac{f'(x)}{g'(x)}$$
>
> [a] `https://de.serlo.org/mathe/funktionen/grenzwerte-stetigkeit-differenzierbarkeit/grenzwert/regel-l-hospital`

Lösungsvorschlag

Es gilt Aussage (b), da $f(n) \in \mathcal{O}(n^3)$ und $g(n) \in \mathcal{O}(n^2)$ und der Grenzwert lim bei größer werdendem n gegen 0 geht. Damit wächst $f(n)$ stärker als $g(n)$, sodass nur Aussage (b) gilt und nicht (a). Dafür nutzen wir die formale Definition des $\mathcal{O}$-Kalküls, indem wir den Grenzwert $\frac{f}{g}$ bzw. $\frac{g}{f}$ bilden:

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = \lim_{n \to \infty} \frac{(n-1)^3}{(2n+3)(3n+2)} = \lim_{n \to \infty} \frac{3(n-1)^2}{(2n+3) \cdot 3 + 2 \cdot (3n+2)} = \lim_{n \to \infty} \frac{6(n-1)}{12} = \infty$$

$$\lim_{n \to \infty} \frac{g(n)}{f(n)} = \lim_{n \to \infty} \frac{(2n+3)(3n+2)}{(n-1)^3} = \lim_{n \to \infty} \frac{(2n+3) \cdot 3 + 2 \cdot (3n+2)}{3(n-1)^2} = \lim_{n \to \infty} \frac{12}{6(n-1)} = 0$$

Hinweis: Hierbei haben wir bei die Regel von L'Hospital angewendet.

Aufgabe 7 [3,5,1,2,4 in leerer Suchbaum und Heap]

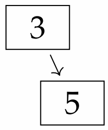
(a) Fügen Sie nacheinander die Zahlen 3, 5, 1, 2, 4

(i) in einen leeren binären Suchbaum ein

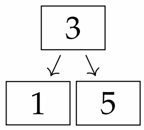
Lösungsvorschlag

Nach dem Einfügen von „3“:

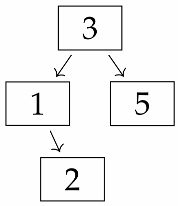
Nach dem Einfügen von „5“:



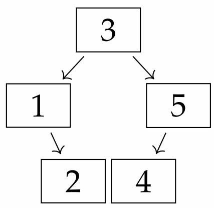
Nach dem Einfügen von „1“:



Nach dem Einfügen von „2“:



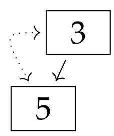
Nach dem Einfügen von „4“:



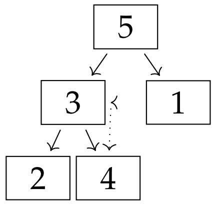
(ii) in einen leeren Heap ein

Lösungsvorschlag

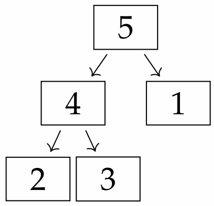
Erstellen einer Max.-Halde, einfügen von 3 und 5, Versickern notwendig:



Einfügen von 1 und 2 ohne Änderungen, Einfügen von 4, versickern notwendig:



Fertiger Heap:



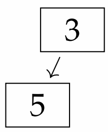
Ausführlicher als Max-Halde

Nach dem Einfügen von „3“:

0
3

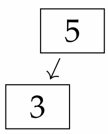
Nach dem Einfügen von „5“:

0	**1**
3	5



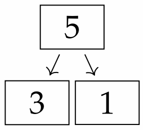
Nach dem Vertauschen von „5“ und „3“:

0	**1**
5	3



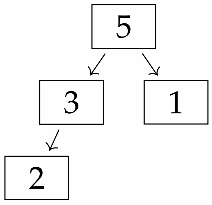
Nach dem Einfügen von „1“:

0	**1**	**2**
5	3	1



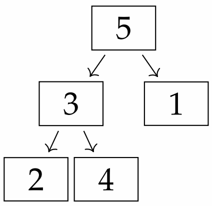
Nach dem Einfügen von „2“:

0	**1**	**2**	**3**
5	3	1	2



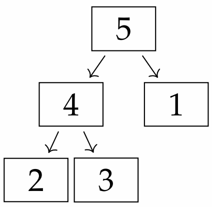
Nach dem Einfügen von „4“:

0	**1**	**2**	**3**	**4**
5	3	1	2	4



Nach dem Vertauschen von „4“ und „3“:

0	**1**	**2**	**3**	**4**
5	4	1	2	3



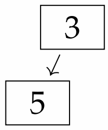
Ausführlicher als Min-Halde

Nach dem Einfügen von „3“:

0
3

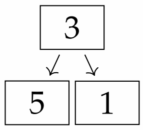
Nach dem Einfügen von „5“:

0	**1**
3	5



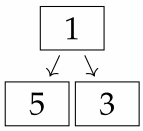
Nach dem Einfügen von „1“:

0	**1**	**2**
3	5	1



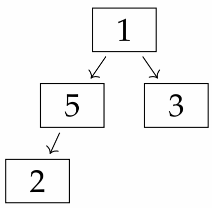
Nach dem Vertauschen von „1“ und „3“:

0	**1**	**2**
1	5	3



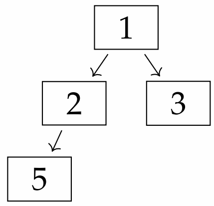
Nach dem Einfügen von „2“:

0	**1**	**2**	**3**
1	5	3	2



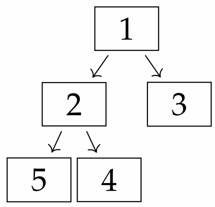
Nach dem Vertauschen von „2“ und „5“:

0	1	2	3
1	2	3	5



Nach dem Einfügen von „4“:

0	1	2	3	4
1	2	3	5	4



Geben Sie die Ergebnisse an (Zeichnung)

(b) Geben Sie zwei Merkmale an, bei denen sich Heaps und binäre Suchbäume wesentlich unterscheiden. Ein wesentlicher Unterschied zwischen Bubblesort und Mergesort ist z. B. die *worst case* Laufzeit mit $\mathcal{O}(n^2)$ für Bubblesort und $\mathcal{O}(n \log n)$ für Mergesort.

Lösungsvorschlag

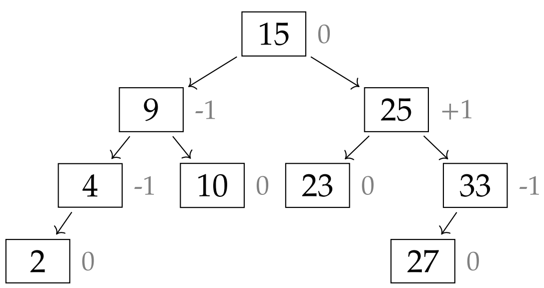
	Binärer Suchbaum	Heap
Suchen beliebiger Wert (worst case)	$\mathcal{O}(\log(n))$	$\mathcal{O}(n)$
Suchen Min-Max (average case)	$\mathcal{O}(\log(n))$	$\mathcal{O}(1)$

[a]

[a] `https://cs.stackexchange.com/q/27860`

Aufgabe 8 [AVL 15,9,25,4,10,23,33,2,27; Einfüge 1,28; Löschen 15]

Gegeben sei der folgende AVL-Baum T. Führen Sie auf T folgende Operationen durch.



Lösungsvorschlag

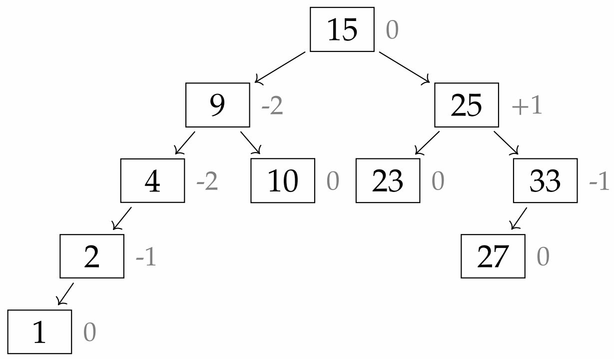
Wir führen alle Operationen am Ursprungsbaum T durch und nicht am veränderten Baum.

(a) Einfüge-Operationen:

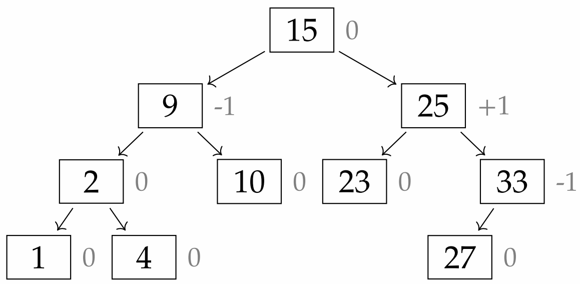
(i) Fügen Sie den Wert 1 in T ein. Balancieren Sie falls nötig und geben Sie den entstandenen Baum (als Zeichnung) an.

Lösungsvorschlag

Nach dem Einfügen von „1“:



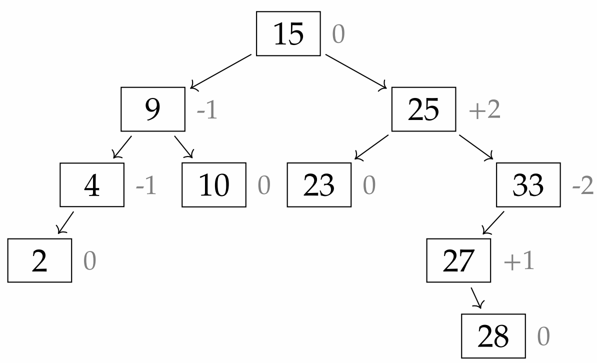
Nach der Rechtsrotation:



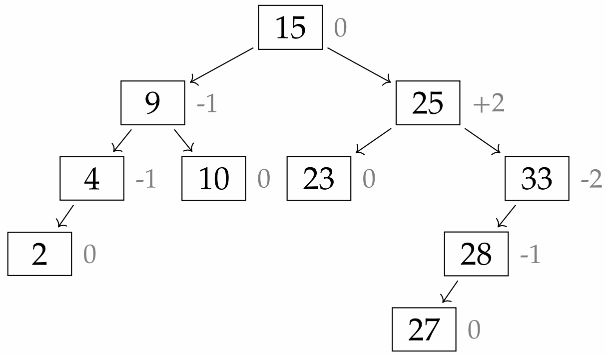
(ii) Fügen Sie nun den Wert 28 in T ein. Balancieren Sie falls nötig und geben Sie den entstandenen Baum (als Zeichnung) an.

Lösungsvorschlag

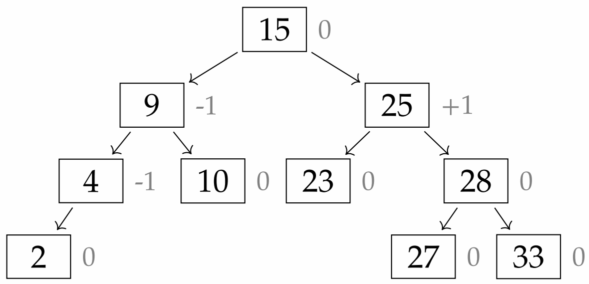
Nach dem Einfügen von „28“:



Nach der Linksrotation:



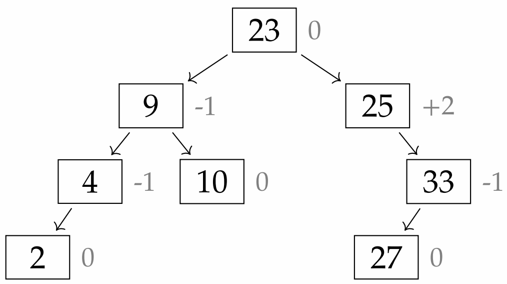
Nach der Rechtsrotation:



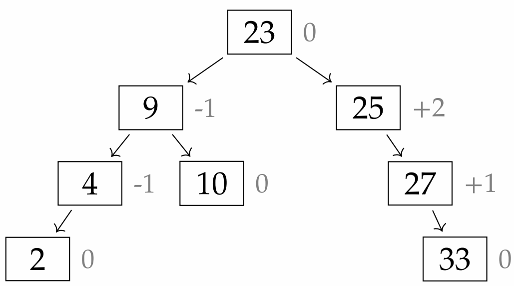
(b) Löschen Sie aus T den Wert 15. Balancieren Sie falls nötig und geben Sie den entstandenen Baum (als Zeichnung) an.

Lösungsvorschlag

Nach dem Löschen von „15":



Nach der Rechtsrotation:



Nach der Linksrotation: