# 66115
# Herbst
# 2013

Theoretische Informatik / Algorithmen (vertieft)

Aufgabenstellungen mit Lösungsvorschlägen

**Die Bschlangaul-Sammlung**

Hermine Bschlangaul and Friends

# Aufgabenübersicht




**Die Bschlangaul-Sammlung**
Hermine Bschlangaul and Friends

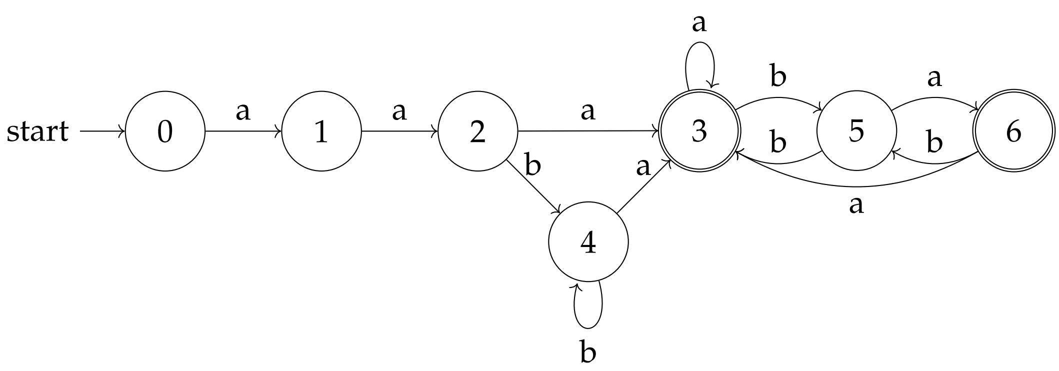
Eine freie Aufgabensammlung mit Lösungen von Studierenden für Studierende zur Vorbereitung auf die 1. Staatsexamensprüfungen des Lehramts Informatik in Bayern.

# Thema Nr. 1

# Thema Nr. 2

**Aufgabe 3 [Minimierung DFA]**

Minimieren Sie den folgenden deterministischen Automaten mit den Zuständen $\{0, 1, 2, 3, 4, 5, 6\}$, dem Startzustand 0 und den Endzuständen $\{3, 6\}$. Geben Sie z. B. durch die Bezeichnung an, welche Zustände zusammengefasst wurden.




Lösungsvorschlag

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ |
| 1 | $x_3$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ |
| 2 | $x_2$ | $x_2$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ |
| 3 | $x_1$ | $x_1$ | $x_1$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ |
| 4 | $x_2$ | $x_2$ | | $x_1$ | $\varnothing$ | $\varnothing$ | $\varnothing$ |
| 5 | $x_2$ | $x_2$ | $x_2$ | $x_1$ | $x_2$ | $\varnothing$ | $\varnothing$ |
| 6 | $x_1$ | $x_1$ | $x_1$ | | $x_1$ | $x_1$ | $\varnothing$ |
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 |

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

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

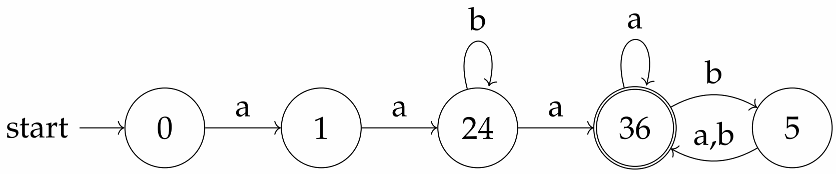
$x_3$ In weiteren Iterationen markierte Zustände.

$x_4$ ...

**Übergangstabelle**

| **Zustandspaar** | **a** | **b** |
|---:|---|---|
| (0, 1) | (1, 2) $x_3$ | (T, T) |
| (0, 2) | (1, 3) $x_2$ | (T, 4) |
| (0, 4) | (1, 3) $x_2$ | (T, 4) |
| (0, 5) | (1, 6) $x_2$ | (T, 3) |
| (1, 2) | (2, 3) $x_2$ | (T, 4) |
| (1, 4) | (2, 3) $x_2$ | (T, 4) |
| (1, 5) | (2, 6) $x_2$ | (T, 3) |
| (2, 4) | (3, 3) | (4, 4) |
| (2, 5) | (3, 6) | (3, 4) $x_2$ |
| (3, 6) | (3, 3) | (5, 5) |
| (4, 5) | (3, 6) | (3, 4) $x_2$ |

T = Trap-Zustand = Falle



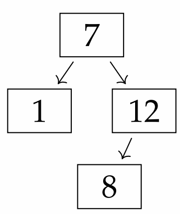

**Aufgabe 7 [Heap und binärer Suchbaum]**

(a)

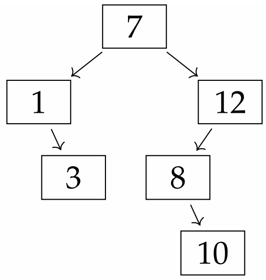
(i) Fügen Sie nacheinander die Zahlen 7, 1, 12, 8, 10, 3, 5 in einen leeren binären Suchbaum ein und zeichnen Sie den Suchbaum nach „8“ und nach „3“.

Lösungsvorschlag

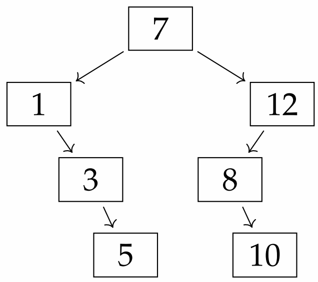
*Nach dem Einfügen von „8“:*




*Nach dem Einfügen von „3“:*



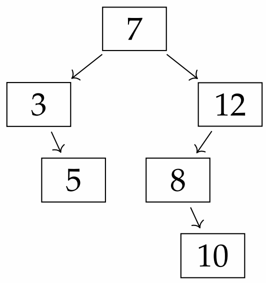
*Nach dem Einfügen von „5“*:



(ii) Löschen Sie die „1“ aus dem in (i) erstellten Suchbaum und zeichnen Sie den Suchbaum.

Lösungsvorschlag

*Nach dem Löschen von „1“*:

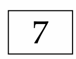


(iii) Fügen Sie 7, 1, 12, 8, 10, 3, 5 in einen leeren MIN-Heap ein, der bzgl. „$\leq$“ angeordnet ist. Geben Sie den Heap nach jedem Element an.

Lösungsvorschlag

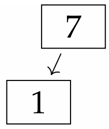
*Nach dem Einfügen von „7“*:

| **0** |
|---|
| 7 |



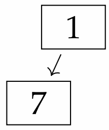
*Nach dem Einfügen von „1“*:

| **0** | **1** |
|---|---|
| 7 | 1 |



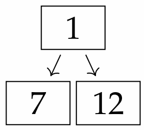
*Nach dem Vertauschen von „1“ und „7“:*

| **0** | **1** |
|---|---|
| 1 | 7 |



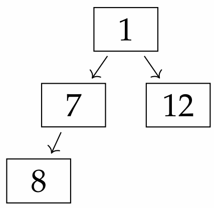
*Nach dem Einfügen von „12“:*

| **0** | **1** | **2** |
|---|---|---|
| 1 | 7 | 12 |



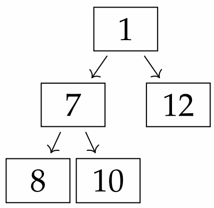
*Nach dem Einfügen von „8“:*

| **0** | **1** | **2** | **3** |
|---|---|---|---|
| 1 | 7 | 12 | 8 |



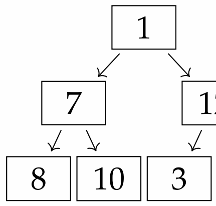
*Nach dem Einfügen von „10“:*

| **0** | **1** | **2** | **3** | **4** |
|---|---|---|---|---|
| 1 | 7 | 12 | 8 | 10 |



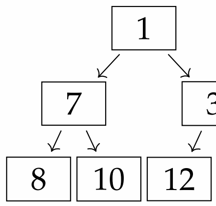
*Nach dem Einfügen von „3“:*

| 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 7 | 12 | 8 | 10 | 3 |



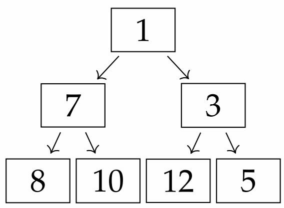

*Nach dem Vertauschen von „3“ und „12“*:

| 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 7 | 3 | 8 | 10 | 12 |




*Nach dem Einfügen von „5“*:

| 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 7 | 3 | 8 | 10 | 12 | 5 |



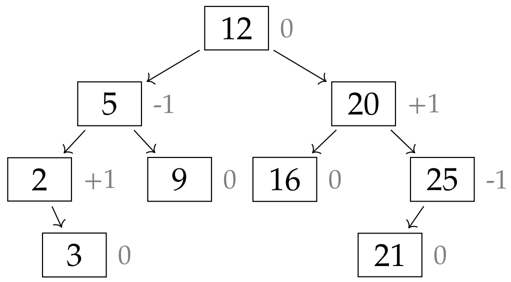

(b) Was ist die worst-case Laufzeit in O-Notation für das Einfügen eines Elements in einen Heap der Größe $n$? Begründen Sie ihre Antwort.

Lösungsvorschlag

Die worst-case Laufzeit berechnet sich aus dem Aufwand für das Durchsickern eines eingefügten Elementes. Da das Durchsickern entlang eines Pfades im Baum erfolgt, entspricht der Aufwand im ungünstigsten Fall der Höhe des Baumes, ð$\log_2 n$. Insgesamt ergibt sich somit eine worst-case Laufzeit von $\mathcal{O}(\log n)$.

**Aufgabe 8 [AVL-Baum 12,5,20,2,9,16,25,3,21]**

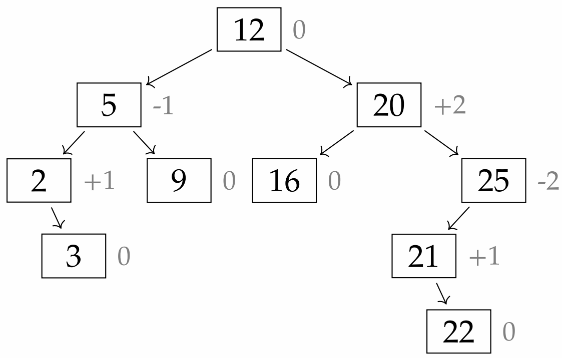
Gegeben sei der folgende AVL-Baum $T$. Führen Sie auf $T$ folgende Operationen durch.



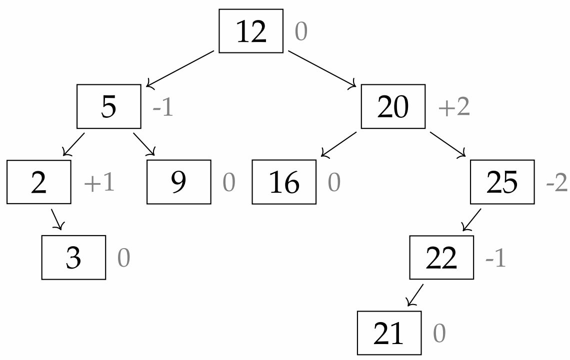

(a) Fügen Sie den Wert 22 in $T$ ein. Balancieren Sie falls nötig und geben Sie den entstandenen Baum (als Zeichnung) an.

Lösungsvorschlag

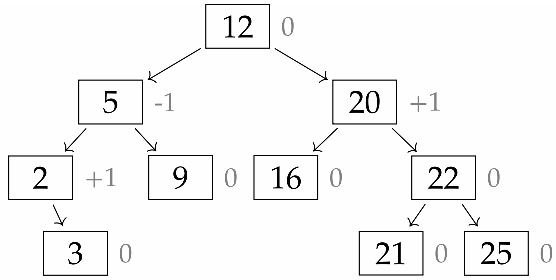
*Nach dem Einfügen von „22“*:




*Nach der Linksrotation*:



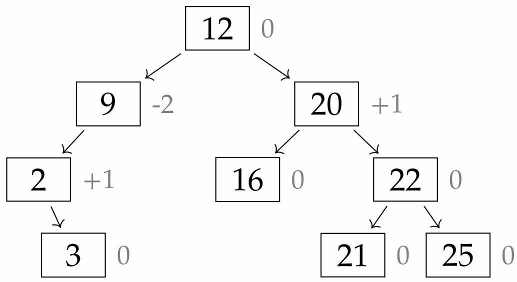

*Nach der Rechtsrotation*:



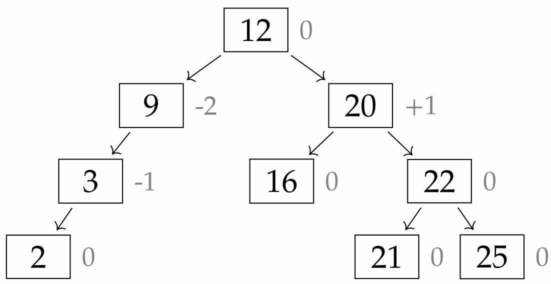

(b) Löschen Sie danach die 5. Balancieren Sie $T$ falls nötig und geben Sie den entstandenen Baum (als Zeichnung) an.

Lösungsvorschlag

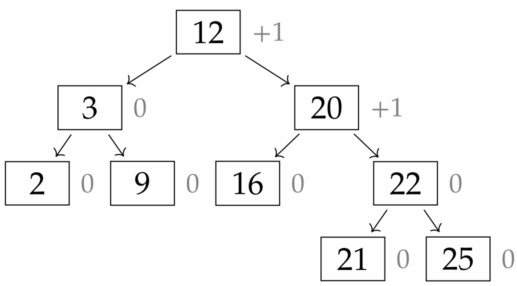
*Nach dem Löschen von „5“*:




*Nach der Linksrotation*:

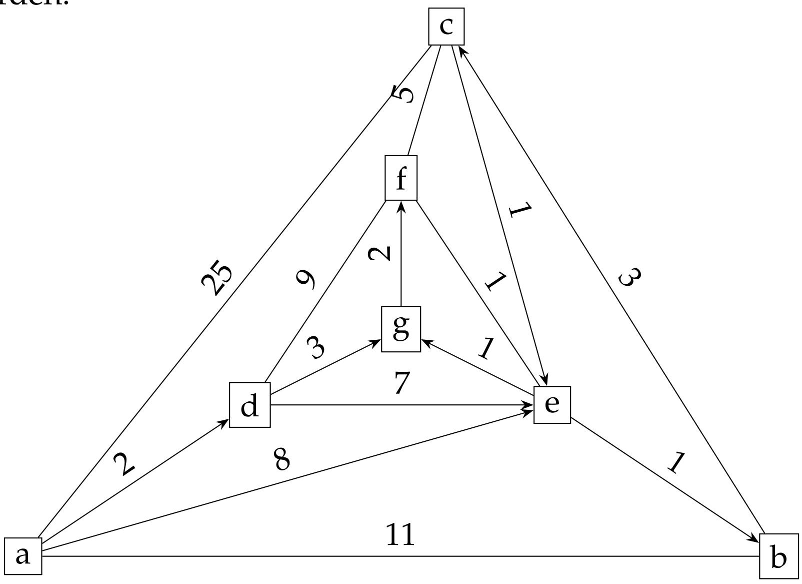



*Nach der Rechtsrotation*:




## 9. Aufgabe [Graph a-g]

Gegeben sei der unten stehende gerichtete Graph $G = (V, E)$ mit positiven Kantenlingen $l(e)$ für jede Kante $e \in E$. Kanten mit Doppelspitzen können in beide Richtungen durchlaufen werden.




(a) In welcher Reihenfolge werden die Knoten von $G$ ab dem Knoten $a$ durch den Dijkstra-Algorithmus bei der Berechnung der kürzesten Wege endgültig bearbeitet?

Lösungsvorschlag

| **Nr.** | **besucht** | **a** | **b** | **c** | **d** | **e** | **f** | **g** |
|---|---|---|---|---|---|---|---|---|
| 0 | | 0 | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| 1 | a | **0** | 11 | 25 | 2 | 8 | ∞ | ∞ |
| 2 | d | \| | 11 | 25 | **2** | 8 | 11 | 5 |
| 3 | g | \| | 11 | 25 | \| | 8 | 7 | **5** |
| 4 | f | \| | 11 | 12 | \| | 8 | **7** | \| |
| 5 | e | \| | 9 | 12 | \| | **8** | \| | \| |
| 6 | b | \| | **9** | 12 | \| | \| | \| | \| |
| 7 | c | \| | \| | **12** | \| | \| | \| | \| |

(b) Berechnen Sie die Länge des kürzesten Weges von $a$ zu jedem Knoten.

Lösungsvorschlag

siehe oben

(c) Geben Sie einen kürzesten Weg von $a$ nach $c$ an.

Lösungsvorschlag

a → d → g → f → c