

Aufgabe 1 Zirkusalgebra

26 Punkte

In der Aufgabe soll ein Zirkus modelliert werden. Ein Zirkus besitzt Tiere, die eindeutig über ihren Namen identifiziert werden können. Zur Vereinfachung der Aufgabe verzichten wir auf die Unterscheidung verschiedener Tierarten, d.h. wir halten die Bändigung einer Maus für genauso spannend wie die eines sibirischen Tigers.

Im Fundus des Zirkusses befinden sich Gegenstände (Kostüme, Requisiten, Aufbauten etc.), welche für die Zirkusnummern gebraucht werden.

Zu einem Zirkus gehört außerdem genau ein Programm, das aus einer Menge von Zirkusnummern besteht.

Zirkusnummern können entweder Artistik-, Clownerie- oder Domptagenummern sein. Für jede dieser Nummern braucht man Gegenstände aus dem Fundus. Da die Akteure nicht nackt auftreten, wird mindestens ein Gegenstand (Kostüm) für eine Nummer benötigt. Jeder Gegenstand hat eine eindeutige Bezeichnung. Eine Domptage erfordert die Beteiligung von mindestens einem Tier. Tiere sind auch bei den anderen Arten von Nummern möglich, jedoch nicht erforderlich.

Jede Zirkusnummer hat eine gewisse Länge in Minuten. Ein Programm darf maximal 120 Minuten lang sein. Umbaupausen sind bereits in den Längenangaben der Nummern enthalten.

Es sollen folgende Operationen spezifiziert werden:

neuerZirkus Es wird ein neuer Zirkus mit einem gegebenen Namen erstellt. Der Fundus und der Tierbestand sind leer. Auch verfügt der Zirkus noch über kein Programm (leer).

ergänzeFundus Hiermit wird der Fundus eines Zirkusses um einen Gegenstand erweitert. Sollte der Gegenstand bereits Teil des Zirkusses sein, so bleibt dessen Fundus unverändert.

kaufeTier Der Tierbestand eines Zirkusses wird um ein Tier erhöht. Auch hier ändert sich am Zirkus nichts, falls bereits ein Tier mit diesem Namen existiert.

erstelleProgramm Erstellt ein leeres Programm.

neueNummer Erweitert ein Programm um eine Zirkusnummer. Würde durch das Einfügen die maximale Länge des Programms überschritten, so bleibt das Programm in seiner ursprünglichen Form erhalten.

länge Ermittelt die Länge eines Programms in Minuten.

übernehmeProgramm Einem Zirkus wird ein Programm zugewiesen. Sind nicht alle notwendigen Mittel (Tiere, Gegenstände) vorhanden, so bleibt das bisherige Programm des Zirkusses bestehen.

(a) Geben Sie eine Signatur für die Algebra *Zirkus* an.

9 Punkte

(b) Spezifizieren Sie geeignete Trägermengen.

7 Punkte

(c) Geben Sie Funktionen für die Operationen an.

10 Punkte

HINWEISE:

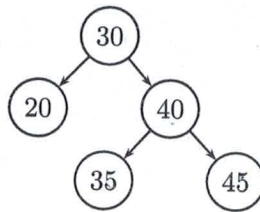
- Trägermengen für *int*, *bool*, *real* und *string* müssen nicht angegeben werden.
- Zur vereinfachten Darstellung können Sie Hilfsmengen und auch Hilfsfunktionen einsetzen, solange diese ordentlich spezifiziert werden.

Aufgabe 2 AVL-Baum

20 Punkte

(a) Gegeben sei der folgende AVL-Baum:

13 Punkte



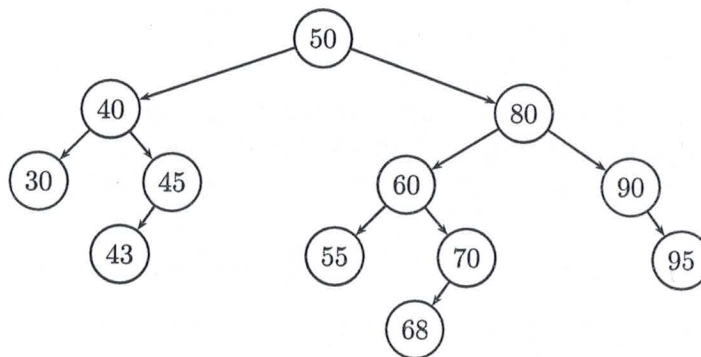
Fügen Sie in diesen AVL-Baum die folgenden Elemente in der gegebenen Reihenfolge ein. Zeichnen Sie den Baum vor und nach einer Strukturänderung und markieren Sie den Knoten, an dem die Balance verletzt ist. Geben Sie weiterhin die durchgeführte Rebalancierungsoperation (Rotation, Doppelrotation) an.

37, 50, 60, 47, 80

Hinweis: Anstatt den Baum vor der Strukturänderung komplett neu zu zeichnen, können Sie auch den letzten rebalancierten Baum erweitern, sofern die neuen Knoten erkennbar sind.

(b) Löschen Sie aus dem folgenden AVL-Baum nacheinander die Knoten 30 und 55. Geben Sie den Baum vor und nach einer Strukturänderung an und markieren Sie den Knoten, an dem die Balance verletzt ist.

7 Punkte

**Aufgabe 3 Sortieren**

28 Punkte

(a) Gegeben sei die folgende Zahlenfolge

14 Punkte

3, 2, 42, 11, 7, 12, 15, 24, 1

Sortieren Sie die obige Zahlenfolge aufsteigend mit Hilfe von Quicksort. Geben Sie dazu den Baum der rekursiven Aufrufe an. Jeder Knoten des Baums entspricht dabei einem Aufruf von Quicksort und enthält die Eingabefolge des Aufrufs. Die Kinder eines Knotens entsprechen den beiden rekursiven Aufrufen. Der Merge-Schritt wird nicht dargestellt. Markieren Sie in jedem Knoten das Pivotelement, das mit der Funktion findx des Kurstextes zu bestimmen ist.



Gegeben sei wieder die Zahlenfolge

3, 2, 42, 11, 7, 12, 15, 24, 1

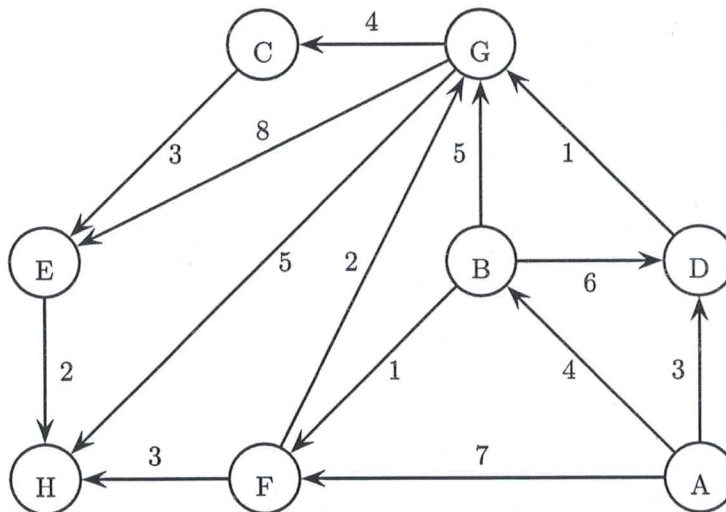
Sortieren Sie diese Folge jetzt absteigend mittels Standard-Heapsort. Geben Sie den initialen Heap an. Stellen Sie weiterhin den Heap und die sortierte Folge nach jedem Einsinkenlassen eines Elements dar. Sie können den Heap wahlweise im Array oder als Baum darstellen.

14 Punkte

Aufgabe 4 Dijkstra

25 Punkte

Gegeben sei folgender Graph:



Bestimmen Sie mit Hilfe des Algorithmus von Dijkstra die kürzesten Wege vom Knoten A zu allen anderen Knoten. Sie können die Lösung zu dieser Aufgabe wahlweise graphisch, textuell oder tabellarisch angeben. Für jeden Schritt des Algorithmus muss erkennbar sein,

- welcher Knoten grün gefärbt wird,
- welche Knoten gelb gefärbt werden und welche Kosten zugewiesen werden,
- welche Kanten rot gefärbt werden,
- welche Kanten gelb gefärbt werden und
- welche neuen Kosten Knoten zugewiesen werden.

Falls unter den gelben Knoten mindestens zwei mit den gleichen Kosten auftreten, so sind diese stets in alphabetisch aufsteigender Reihenfolge zu verarbeiten.

Aufgabe 5 Deckblatt

1 Punkt

Lesen Sie sich die „Hinweise zur Bearbeitung“ sorgfältig durch. Füllen Sie den dort aufgeführten Anweisungen entsprechend beide Deckblätter vollständig und korrekt aus.



FernUniversität in Hagen

**Lösungsvorschläge
zur Hauptklausur
1661 „Datenstrukturen I“**

13.08.2016

Aufgabe 1 Zirkusalgebra

(a)

algebra *zirkus*

sorts *zirkus, nummer, programm, time*

ops

<i>neuerZirkus</i> :	<i>string</i>	→	<i>zirkus</i>
<i>ergänzeFundus</i> :	<i>zirkus</i> × <i>string</i>	→	<i>zirkus</i>
<i>kaufeTier</i> :	<i>zirkus</i> × <i>string</i>	→	<i>zirkus</i>
<i>erstelleProgramm</i>		→	<i>programm</i>
<i>neueNummer</i> :	<i>programm</i> × <i>nummer</i>	→	<i>programm</i>
<i>länge</i> :	<i>programm</i>	→	<i>time</i>
<i>übernehmeProgramm</i> :	<i>zirkus</i> × <i>programm</i>	→	<i>zirkus</i>

(b)

Zunächst definieren wir uns eine Hilfsmenge *stringset*, die eine endliche Menge von Strings darstellt, wie folgt:

$$\text{stringset} = \{S \subseteq \text{string} \mid S \text{ ist endlich}\}$$

Weiterhin definieren wir uns eine Menge *Typ*, die zur Unterscheidung der verschiedenen Arten von Nummern verwendet wird:

$$\text{Typ} = \{\text{art}, \text{dom}, \text{clo}\}$$

sets

<i>time</i>	=	$\{t \in \text{int} \mid t > 0\}$
<i>nummer</i>	=	$\{(A, G, T, Z) \in \text{Type} \times \text{stringset} \times \text{stringset} \times \text{time} \mid G \neq \emptyset \wedge (T \neq \emptyset \vee A \neq \text{dom})\}$
<i>programm</i>	=	$\{P \subseteq \text{nummer} \mid \text{länge}(P) \leq 120\}$
<i>zirkus</i>	=	$\{(N, T, F, P) \in \text{string} \times \text{stringset} \times \text{stringset} \times \text{programm} \mid \text{getTiere}(P) \setminus T = \emptyset \wedge \text{getFundus}(P) \setminus F = \emptyset\}$

Hierbei haben wir folgende Hilfsfunktionen benutzt:

$$\begin{aligned} \text{getTiere} &: \text{programm} \rightarrow \text{stringset} \\ \text{getTiere}(P) &= \bigcup_{(a,g,t,z) \in P} t \\ \text{getFundus} &: \text{programm} \rightarrow \text{stringset} \\ \text{getFundus}(P) &= \bigcup_{(a,g,t,z) \in P} g \end{aligned}$$

(c)

functions

$$\text{neuerZirkus}(n) = (n, \emptyset, \emptyset, \emptyset)$$

$$\text{ergänzeFundus}((n, ti, fu, pro), bez) = (n, ti, fu \cup bez, pro)$$

$$\text{kaufeTier}((n, tie, fu, pro), tname) = (n, tie \cup tname, fu, pro)$$

$$\text{erstelleProgramm}() = \emptyset$$

$$\text{neueNummer}(P, (a, g, t, z)) = \begin{cases} P & \text{falls } \text{länge}(P) + z > 120 \\ P \cup \{(a, g, t, z)\} & \text{sonst} \end{cases}$$

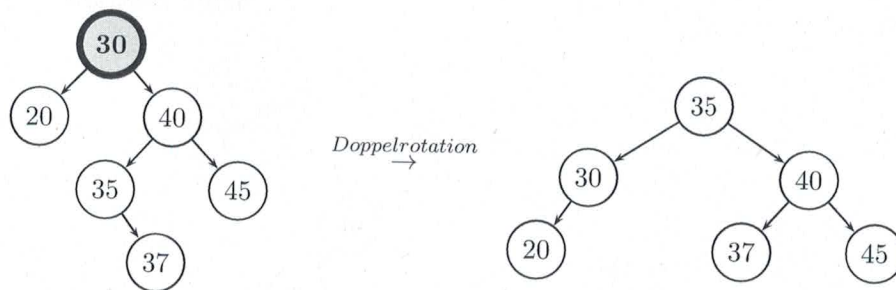
$$\text{länge}(P) = \sum_{(a,g,t,z) \in P} z$$

$$\text{übernehmeProgramm}((N, T, F, PZ), P) = \begin{cases} (N, T, F, P) & \text{falls } \text{getTiere}(P) \setminus T = \emptyset \wedge \\ & \text{getFundus}(P) \setminus F = \emptyset \\ (N, T, F, PZ) & \text{sonst} \end{cases}$$

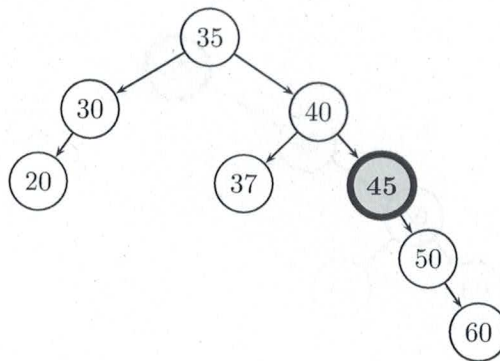
Aufgabe 2 AVL-Baum

(a)

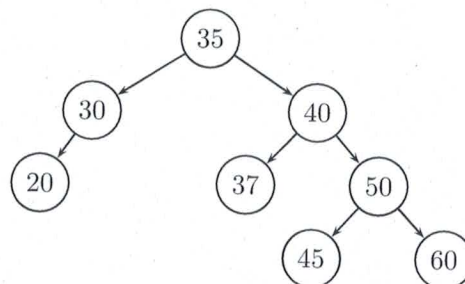
Das Einfügen von 37 führt zu einer Verletzung der Balance am markierten Knoten:



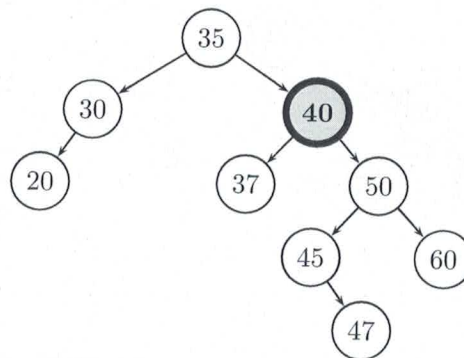
Das Einfügen von 60 führt zu einer Verletzung der Balance am markierten Knoten:



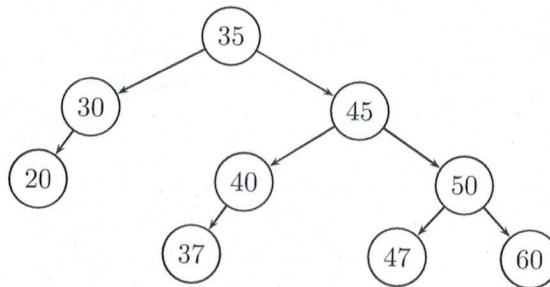
Wir führen eine Linksrotation durch:



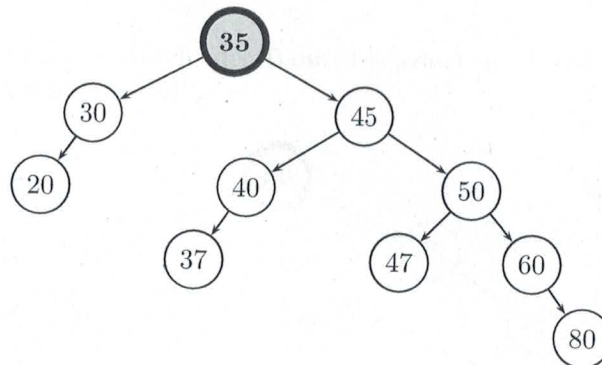
Das Einfügen von 47 führt zu einer Verletzung der Balance am markierten Knoten:



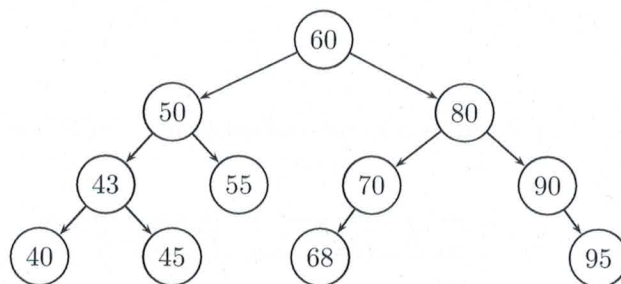
Wir führen eine Doppelrotation durch:



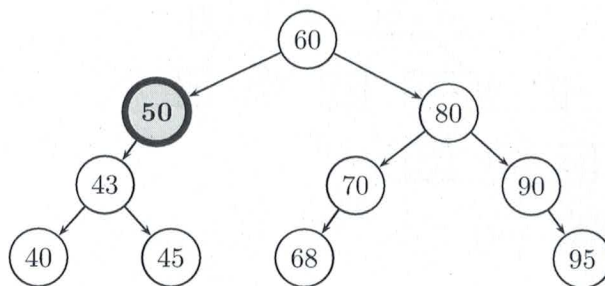
Das Einfügen von 80 führt zu einer Verletzung der Balance am markierten Knoten:



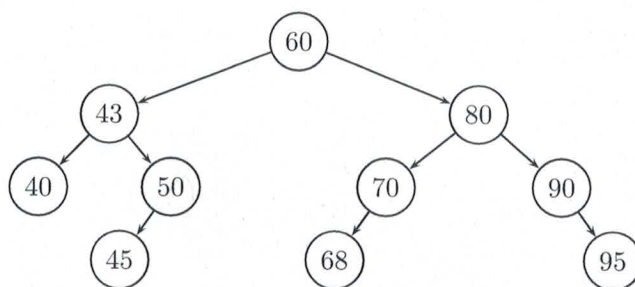
Wiederum muss eine Doppelrotation durchgeführt werden, die zu folgendem Baum führt:



Löschen des Werts 55:



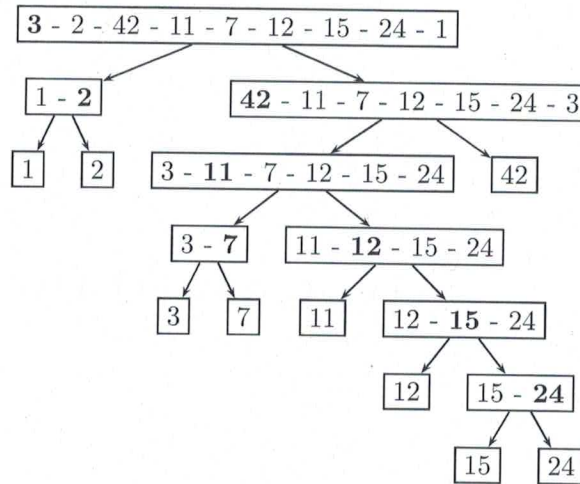
Wir führen eine Rechtsrotation durch:



Aufgabe 3 Sortieren

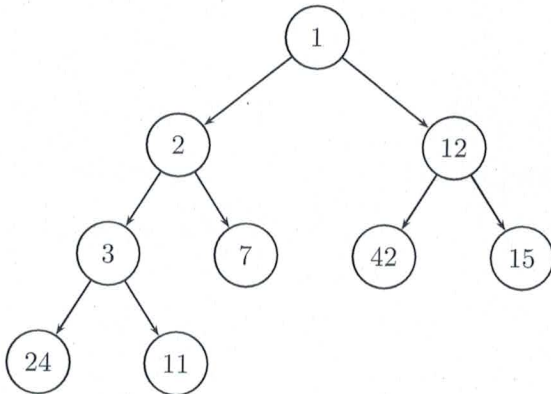
(a)

Der Baum der rekursiven Aufrufe von Quicksort für die in der Aufgabe angegebene Folge sieht wie folgt aus:



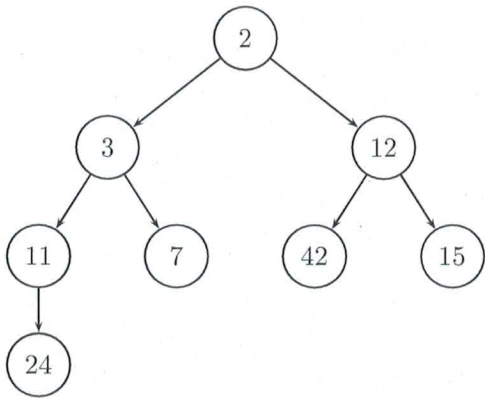
(b)

Der initiale Heap lautet:



Arrayeinbettung : | 1 | 2 | 12 | 3 | 7 | 42 | 15 | 24 | 11 |

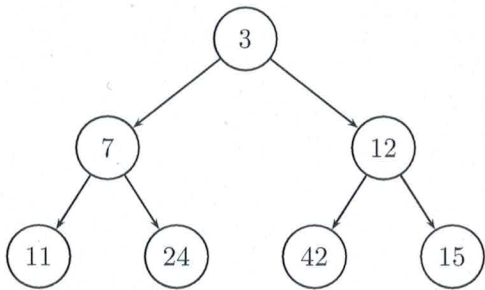
Element 1 verarbeitet



Sortierte Folge: 1

Arrayeinbettung : | 2 | 3 | 12 | 11 | 7 | 42 | 15 | 24 || 1 |

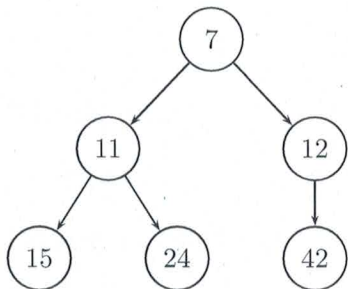
Element 2 verarbeitet



Sortierte Folge: 2, 1

Arrayeinbettung : | 3 | 7 | 12 | 11 | 24 | 42 | 15 || 2 | 1 |

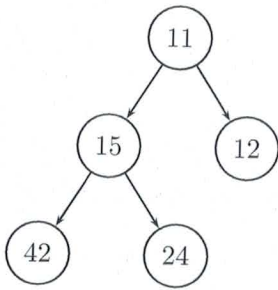
Element 3 verarbeitet



Sortierte Folge: 3, 2, 1

Arrayeinbettung : | 7 | 11 | 12 | 15 | 24 | 42 || 3 | 2 | 1 |

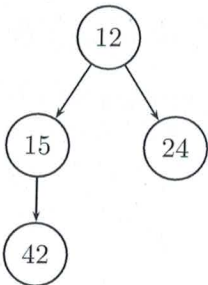
Element 7 verarbeitet



Sortierte Folge: 7, 3, 2, 1

Arrayeinbettung : | 11 | 15 | 12 | 42 | 24 || 7 | 3 | 2 | 1 |

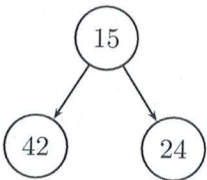
Element 11 verarbeitet



Sortierte Folge: 11, 7, 3, 2, 1

Arrayeinbettung : | 12 | 15 | 24 | 42 || 11 | 7 | 3 | 2 | 1 |

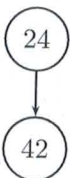
Element 12 verarbeitet



Sortierte Folge: 12, 11, 7, 3, 2, 1

Arrayeinbettung : | 15 | 42 | 24 || 12 | 11 | 7 | 3 | 2 | 1 |

Element 15 verarbeitet



Sortierte Folge: 15, 12, 11, 7, 3, 2, 1

Arrayeinbettung : | 24 | 42 || 15 | 12 | 11 | 7 | 3 | 2 | 1 |

Element 24 verarbeitet

42

Sortierte Folge: 24, 15, 12, 11, 7, 3, 2, 1

Arrayeinbettung : | 42 || 24 | 15 | 12 | 11 | 7 | 3 | 2 | 1 |

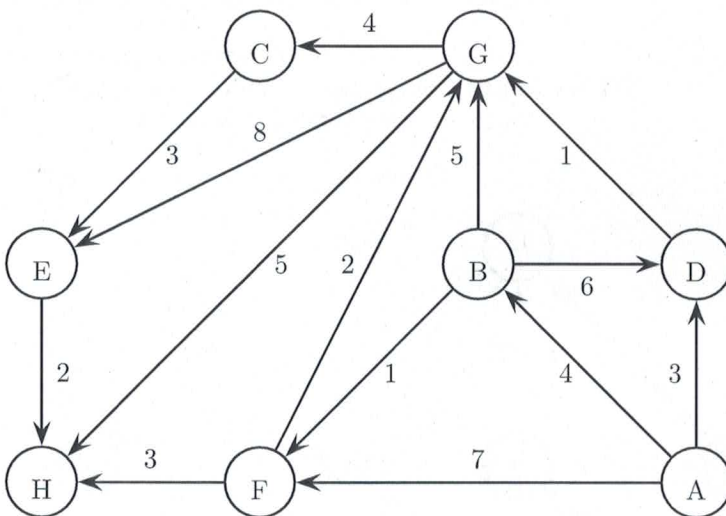
Element 42 verarbeitet

Sortierte Folge: 42, 24, 15, 12, 11, 7, 3, 2, 1

Arrayeinbettung : || 42 | 24 | 15 | 12 | 11 | 7 | 3 | 2 | 1 |

Aufgabe 4 Dijkstra

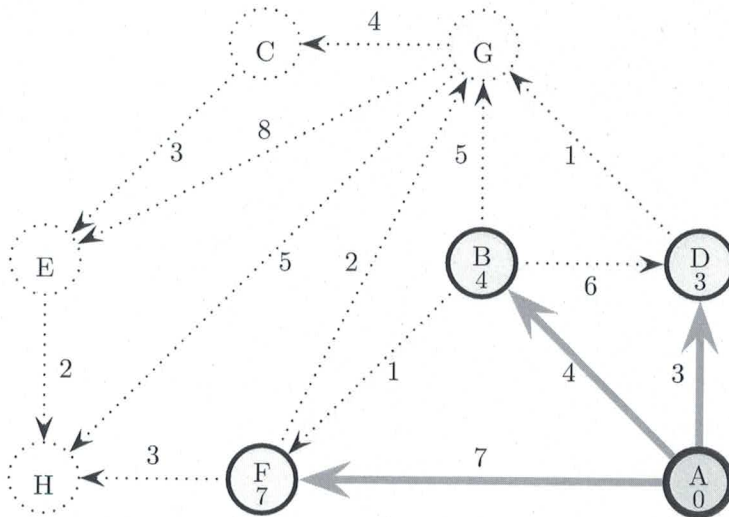
Ausgangsgraph :



Aktionen beim Verarbeiten des Knotens A

- Färbe Knoten A grün
- Färbe Knoten B gelb, setze Distanz auf 4.0
- Färbe Kante (A - B) rot
- Färbe Knoten D gelb, setze Distanz auf 3.0
- Färbe Kante (A - D) rot
- Färbe Knoten F gelb, setze Distanz auf 7.0
- Färbe Kante (A - F) rot

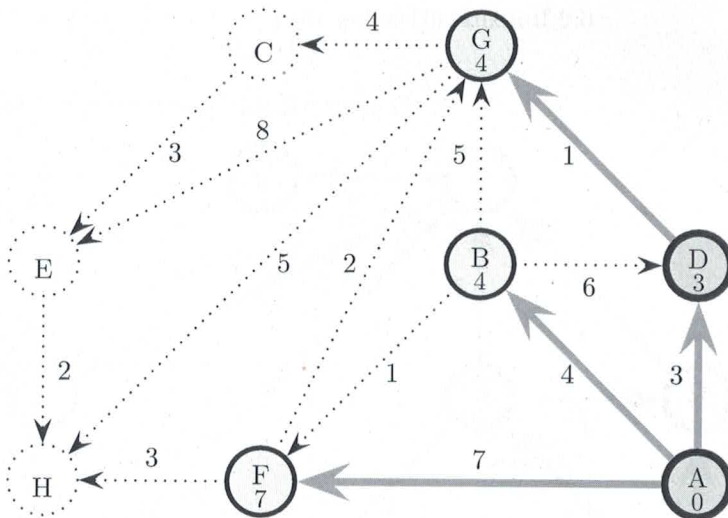
Nach Verarbeitung des Knotens A



Aktionen beim Verarbeiten des Knotens D

- Färbe Knoten D grün
- Färbe Knoten G gelb, setze Distanz auf 4.0
- Färbe Kante (D - G) rot

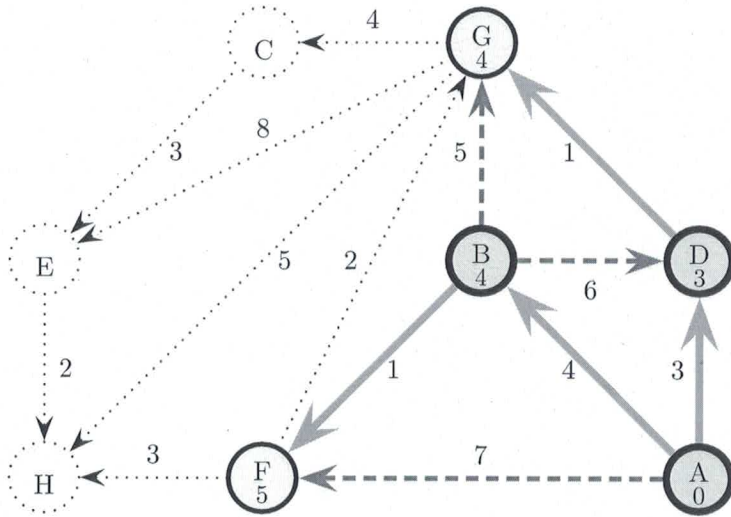
Nach Verarbeitung des Knotens D



Aktionen beim Verarbeiten des Knotens B

- Färbe Knoten B grün
- Färbe Kante (B - D) gelb
- Kürzerer Weg zu Knoten F gefunden, neue Distanz ist:5.0
- Färbe Kante (A F) gelb
- Färbe Kante (B - F) rot
- Färbe Kante (B - G) gelb

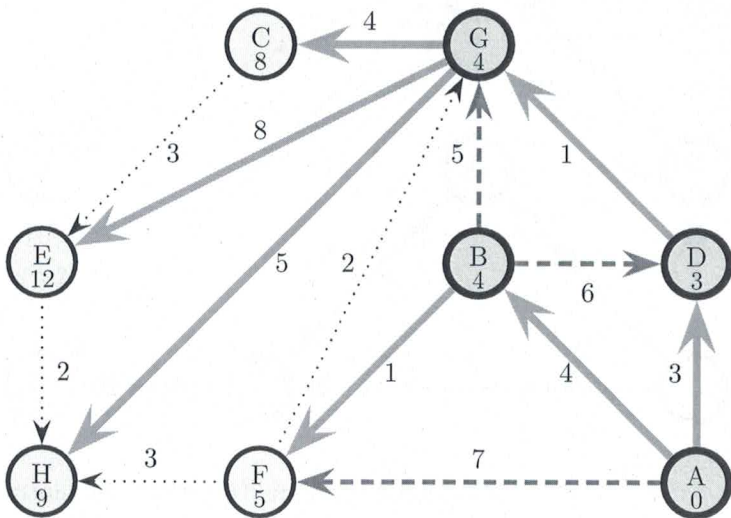
Nach Verarbeitung des Knotens B



Aktionen beim Verarbeiten des Knotens G

- Färbe Knoten G grün
- Färbe Knoten C gelb, setze Distanz auf 8.0
- Färbe Kante (G - C) rot
- Färbe Knoten E gelb, setze Distanz auf 12.0
- Färbe Kante (G - E) rot
- Färbe Knoten H gelb, setze Distanz auf 9.0
- Färbe Kante (G - H) rot

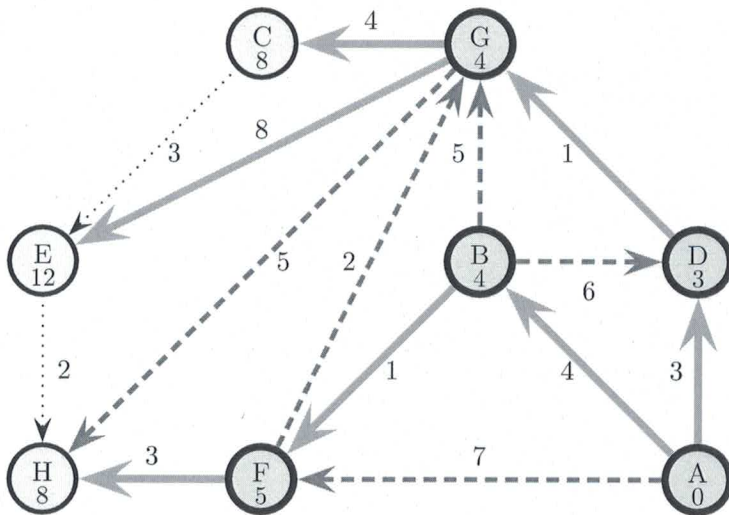
Nach Verarbeitung des Knotens G



Aktionen beim Verarbeiten des Knotens F

- Färbe Knoten F grün
- Färbe Kante (F - G) gelb
- Kürzerer Weg zu Knoten H gefunden, neue Distanz ist:8.0
- Färbe Kante (G - H) gelb
- Färbe Kante (F - H) rot

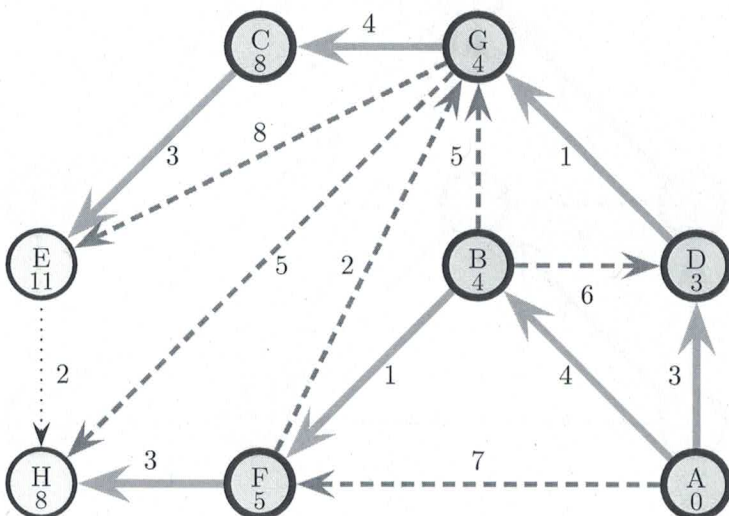
Nach Verarbeitung des Knotens F



Aktionen beim Verarbeiten des Knotens C

- Färbe Knoten C grün
- Kürzerer Weg zu Knoten E gefunden, neue Distanz ist:11.0
- Färbe Kante (G E) gelb
- Färbe Kante (C - E) rot

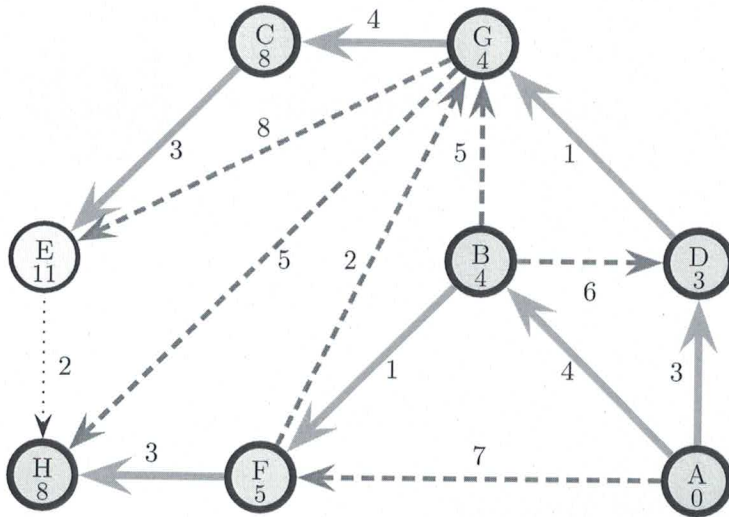
Nach Verarbeitung des Knotens C



Aktionen beim Verarbeiten des Knotens H

- Färbe Knoten H grün

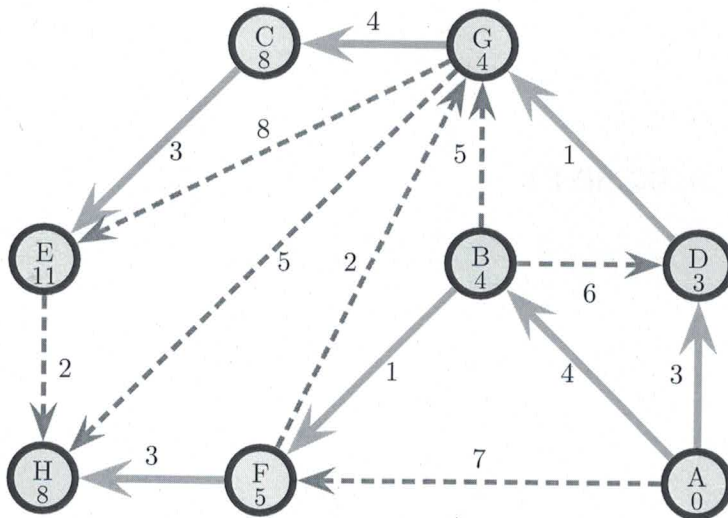
Nach Verarbeitung des Knotens H



Aktionen beim Verarbeiten des Knotens E

- Färbe Knoten E grün
- Färbe Kante (E - H) gelb

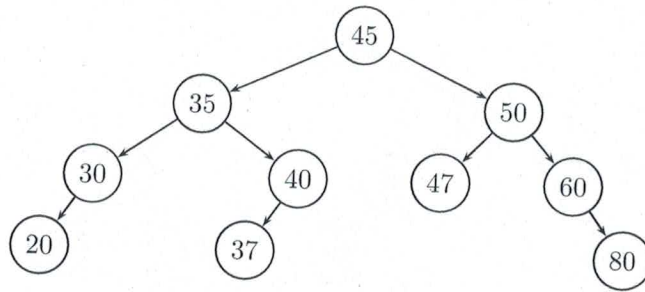
Verarbeitung des letzten Knotens



Aufgabe 5 Deckblatt

Hier erhalten Sie den Punkt, wenn Sie beide Klausurdeckblätter korrekt und vollständig ausgefüllt haben, also Namen, Matrikelnummer und Adresse korrekt eingetragen und genau diejenigen Aufgaben markiert haben, die Sie auch bearbeitet haben.

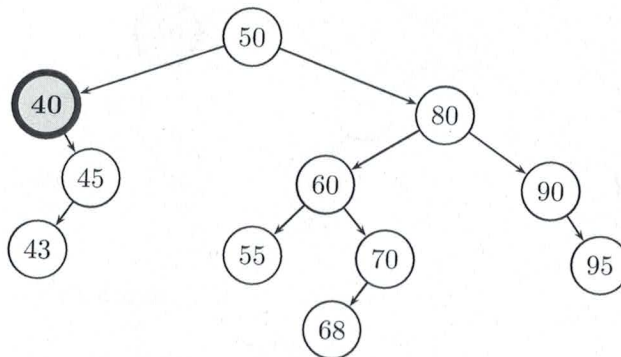
Wir führen eine Linksrotation durch:



Dies stellt gleichzeitig den endgültigen Baum dar.

(b)

Löschen des Werts 30:



Wir führen eine Doppelrotation durch, der Baum ist jedoch noch nicht ausgeglichen, so dass eine weitere Balancierung notwendig wird.

