

Algorithmen und Datenstrukturen
Klausur SS 2018
Angewandte Informatik Bachelor

Name	
Matrikelnummer	

Aufgabe 1	Verständnisfrage	6	
Aufgabe 2	AVL-Baum	14	
Aufgabe 3	Algorithmus von Dijkstra	15	
Aufgabe 4	Tiefensuchbaum und Artikulationspunkte	13	
Aufgabe 5	Binäre und binomiale Heaps	12	
Summe		60	

Aufgabe 1 Verständnisfrage**(6 Punkte)**

Für eine Anwendung wird sowohl das Löschen des kleinsten Elements als auch das Löschen des größten Elements benötigt:

```
DataStructure data = ...;  
// ...  
min = data.delMin();  
max = data.delMax();  
// ...
```

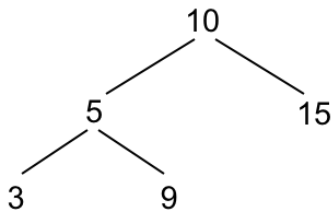
Welche der folgenden Datenstrukturen bietet hierfür effiziente Operationen ($O(\log n)$ oder besser) an. Kennzeichnen Sie mit "+" für ja bzw. "-" für nein.

Datenstruktur	unterstützt sowohl das Löschen des größten als auch das Löschen des kleinsten Elements effizient
Sortiertes Feld	+
Sortierte einfach verkettete Liste	-
AVL-Baum	+
Heap	-
Hashtabelle	-
B-Baum	+

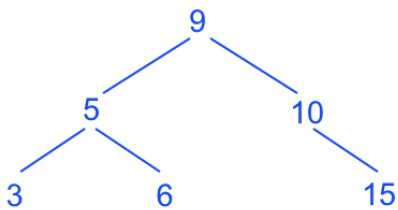
Aufgabe 2 AVL-Baum

(14 Punkte)

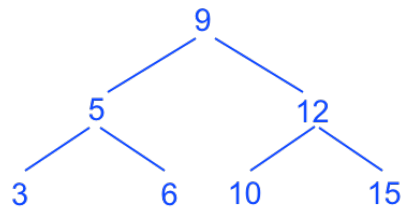
a) Fügen Sie in folgendem AVL-Baum die Zahlen 6 und dann 12 ein:



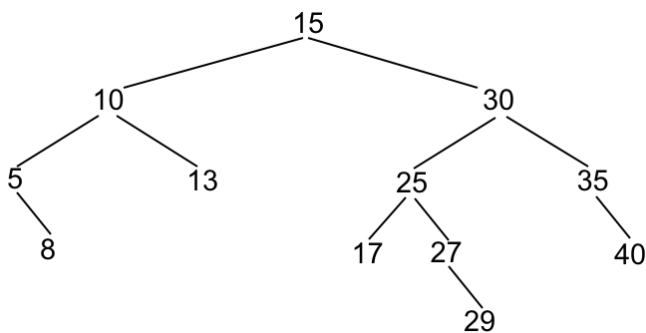
Einfügen: 6



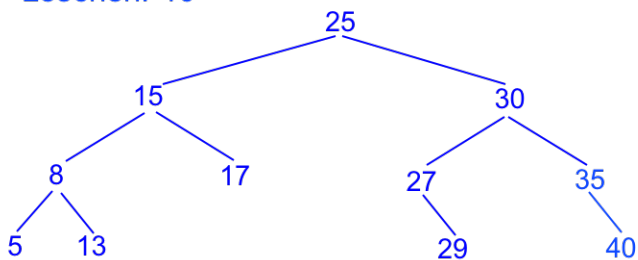
Einfügen: 12



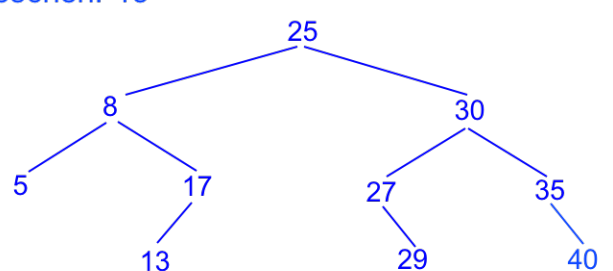
b) Löschen Sie in folgendem AVL-Baum die Zahl 10, 15 und dann 25. Halten Sie dabei die folgende Regel ein: Wird ein Knoten mit zwei Kindern gelöscht, dann wird er durch das Minimum im rechten Teilbaum ersetzt.



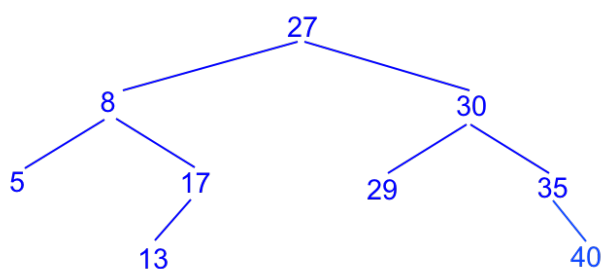
Löschen: 10



Löschen: 15



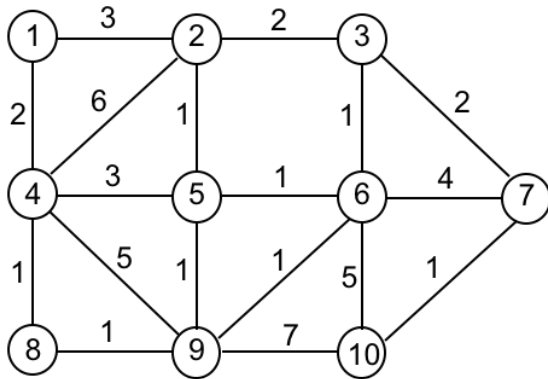
Löschen: 25



Aufgabe 3 Algorithmus von Dijkstra

(15 Punkte)

Gegeben ist ein ungerichteter Graph mit Kosten als Gewichte. Bestimmen Sie mit dem Algorithmus von Dijkstra vom Startknoten $s = 4$ zu allen anderen Knoten jeweils einen günstigsten Weg.



a) Tragen Sie in folgende Tabelle nach jedem Besuchsschritt folgendes ein:

- der besuchte Knoten b
- die Kosten $d[v]$ für den günstigsten Weg von Startknoten s nach v
- den Vorgängerknoten $p[v]$ für den günstigsten Weg von Startknoten s nach v .

Wichtig: Haben mehrere Kandidaten denselben d -Wert, dann wird der Kandidat mit kleinster Nummer als nächster Knoten besucht.

Hinweis: Es brauchen nur die d - und p -Werte eingetragen werden, die sich geändert haben. Die endgültigen p - und d -Werte können durch Umrandung besonders gekennzeichnet werden.

b	d[1]	d[2]	d[3]	d[4]	d[5]	d[6]	d[7]	d[8]	d[9]	d[10]	p[1]	p[2]	p[3]	p[4]	p[5]	p[6]	p[7]	p[8]	p[9]	p[10]
4	2	6	∞	0	3	∞	∞	1	5	∞	4	4	-	-	4	-	-	4	4	
8									2										8	
1		5										1								
9						3			9							9				9
5		4										5								
6			4				7		8				6				6			6
2																				
3							6										3			
7									7											7
10																				

b) Geben Sie den gefundenen günstigsten Weg von 4 nach 10 an.

4 – 8 – 9 – 6 – 3 – 7 – 10

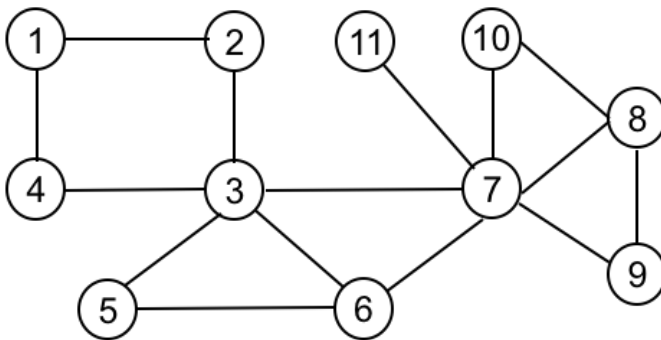
c) Welche Kosten hat der günstigste Weg von 4 nach 10?

7

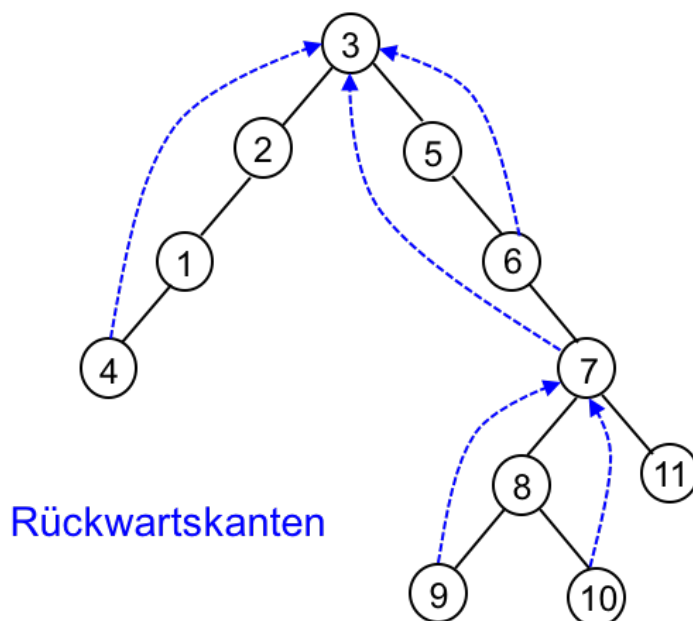
Aufgabe 4 *Tiefensuchbaum und Artikulationspunkte*

(13 Punkte)

Gegeben sei folgender ungerichteter Graph:



- a) Geben Sie den Tiefensuchbaum mit Rückwärtskanten (TR) mit Wurzel 3 an. Betrachten Sie die Nachbarn eines Knotens in der durch die Knotennummerierung gegebenen Reihenfolge.



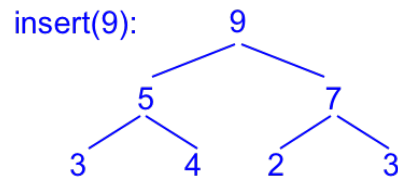
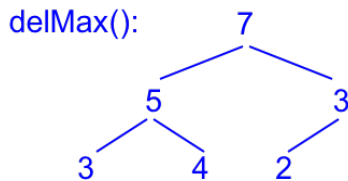
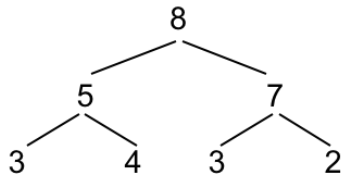
- b) Begründen Sie mit Hilfe des TR, warum Knoten 3 und 7 Artikulationspunkte (APe) und Knoten 8 kein Artikulationspunkt ist? Folgender Begriff darf verwendet werden: Ein Rückwärtsweg ist ein Weg in einem Tiefensuchbaum mit einer beliebig langen Folge von Vorwärtskanten und dann genau einer Rückwärtskante.

- Knoten 3 ist ein AP, da 3 die Wurzel ist und mehr als 1 Kind hat.
- Knoten 7 ist ein AP, da 7 im TR ein Kind hat (z.B. 8), von dem es keinen Rückwärtsweg zu einem Vorfahren von 7 gibt.
- Knoten 8 ist kein AP, da jedes Kind von 8 (nämlich 9 und 10)) einen Rückwärtsweg zu einem Vorfahren von 8 (nämlich 7) hat.

Aufgabe 5 Binäre und binomiale Heaps

(12 Punkte)

- a) Löschen Sie in folgendem binären Heap zuerst die größte Zahl und fügen Sie dann die Zahl 9 ein. Der Heap ist absteigend heap-geordnet: d.h. in jeder Elternknoten ist größer oder gleich als seine Kindknoten.



- b) Löschen Sie in folgendem binomialen Heap zweimal die größte Zahl.

