Datenstrukturen und Algorithmen, SS 2019
Prof. Dr. P. Rossmanith
S. Dollase, J. Dreier, H. Lotze

Datum: 06.09.2019

Klausur mit Lösungen 02

Aufgabe K1 (11 Punkte)

Wir vergleichen verschiedene Datenstrukturen für die Implementierung einer *Menge* von Elementen (ohne Duplikate). Geben Sie eine obere Schranke für die Laufzeiten an. Gehen Sie davon aus, dass die Anzahl der in der Datenstruktur enthaltenden Elemente n beträgt. Tragen Sie eine Funktion $f(n)$ in die Tabelle ein, um eine Worst Case Laufzeit von $O(f(n))$ auszudrücken. Bei randomisierten Datenstrukturen ist die erwartete Laufzeit gemeint. Beachten Sie, dass beim Löschen nur der Schlüssel bekannt ist und nicht ob und an welcher Stelle in der Datenstruktur das Element abgelegt ist (Arrayindex, Listenknoten, ...) und dass beim Einfügen eines Schlüssels dieser schon enthalten sein könnte und dann nicht erneut eingefügt werden darf.

	AVL-Baum	Skip-List	Sortierte Liste	Array	Sortiertes Array	Binärer Suchbaum	Liste	Treap	Min-Heap	Hashtabelle
Suchen										
Einfügen										
Löschen										
Sort. Ausgeben										
Minimum										
Maximum										

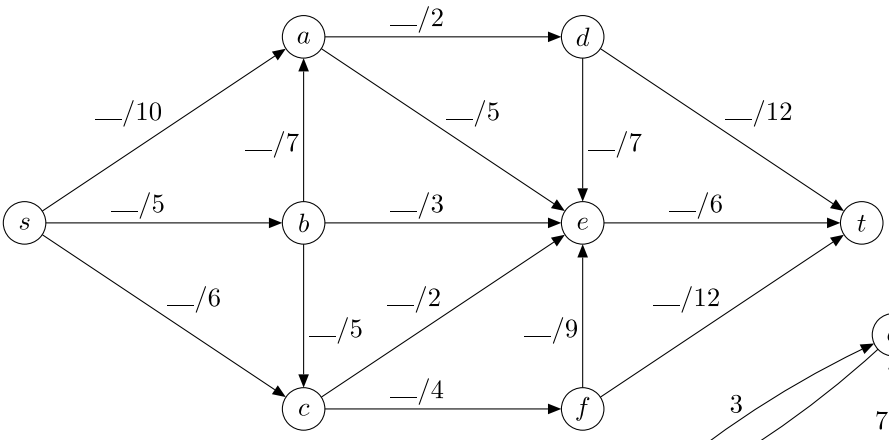
Lösungsvorschlag

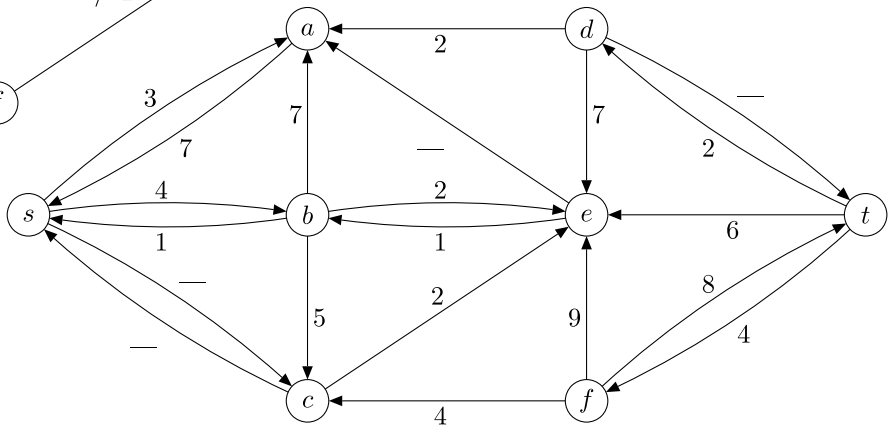
	AVL-Baum	Skip-List	Sortierte Liste	Array	Sortiertes Array	Binärer Suchbaum	Liste	Treap	Min-Heap	Hashtabelle
Suchen	$\log n$	$\log n$	n	n	$\log n$	n	n	$\log n$	n	1
Einfügen	$\log n$	$\log n$	n	n	n	n	n	$\log n$	n	1
Löschen	$\log n$	$\log n$	n	n	n	n	n	$\log n$	n	1
Sort. Aus.	n	n	n	$n \log n$	n	n	$n \log n$	n	$n \log n$	$n \log n$
Minimum	$\log n$	1	1	n	1	n	n	$\log n$	1	n
Maximum	$\log n$	$\log n$	1	n	1	n	n	$\log n$	n	n

Spezialfälle: Minimum und Maximum in Sortierter Liste ggf auch $1, n$ oder $n, 1$, aber nicht n, n.

Aufgabe K2 (8+2+3 Punkte)

a) Sie finden links ein Flussnetzwerk G. Rechts ist das dazugehörige Residualnetzwerk G_f bezüglich eines Flusses f. Zeichnen Sie den Fluss f in die linke Zeichnung auf den Strichen ein und rechts die fehlenden Kapazitäten in G_f.



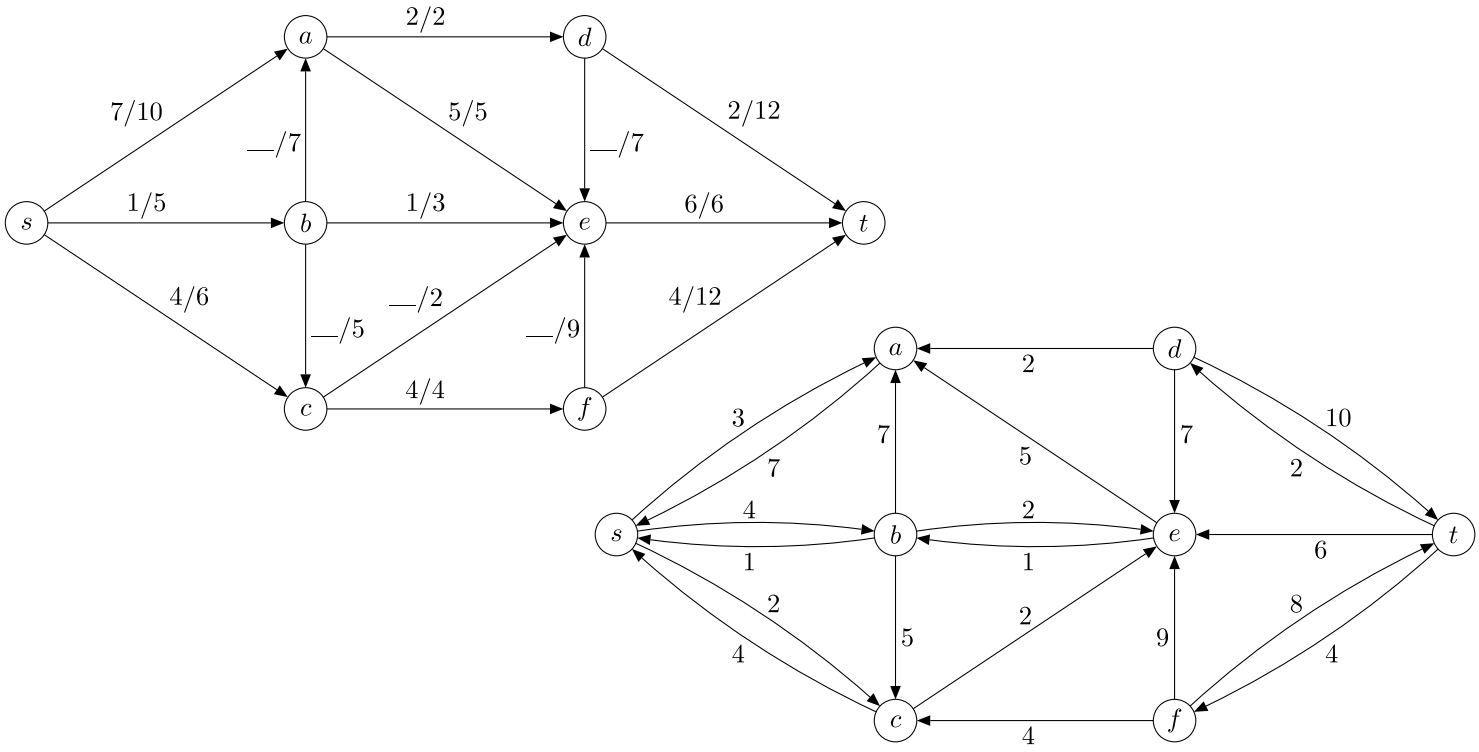


b) Der Wert des Flusses f beträgt $|f| =$ ☐.

c) Ist f maximal? Begründen Sie Ihre Antwort kurz.

Lösungsvorschlag

a)



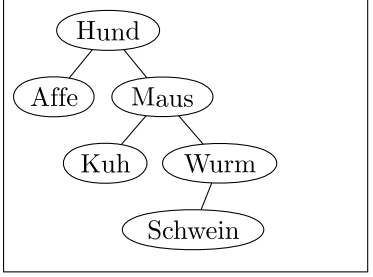
b) Der Wert des Flusses f beträgt $|f| = 12$.

c) Ja, denn es gibt keinen s-t-Pfad im Residualnetzwerk.

Alternativ: Der gefundene Fluss ist maximal, da der Schnitt $(S, T) = (\{s, a, b, c, e\}, \{d, f, t\})$ die Kapazität 12 hat und es somit keinen größeren Fluss geben kann.

Aufgabe K3 (2+2+2+6 Punkte)

Wie sieht folgender Splay-Baum jeweils aus, nachdem eine dieser Operationen ausgeführt worden sind: a) Suche nach „Maus“, b) Löschen von „Kuh“, c) Suche nach „Ratte“. Zeichnen Sie die entsprechenden Splay-Bäume in die Kästen ein. (Die Operationen sollen nicht nacheinander ausgeführt werden sondern immer auf dem ursprünglichen Baum!) Verwenden Sie genau die Algorithmen der Vorlesung. Sie dürfen beim Zeichnen der Splay-Bäume die Knotennamen mit ihrem entsprechenden Anfangsbuchstaben abkürzen, also etwa A für Affe oder S für Schwein.



a)

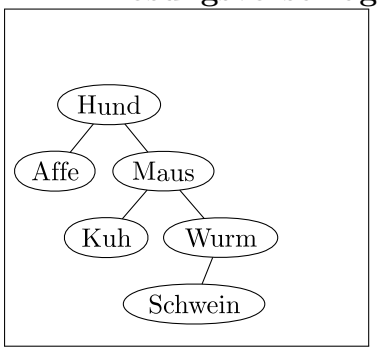
b)

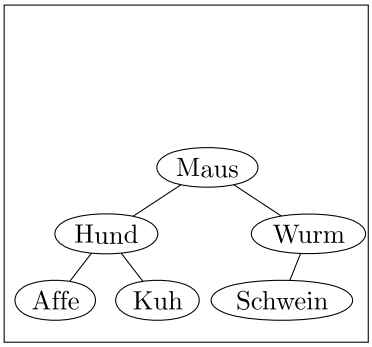
c)

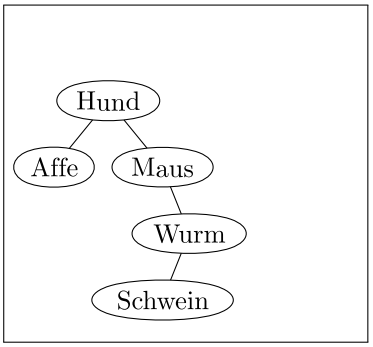
d) Nehmen Sie an, es gäbe einen Splay-Baum, dessen genaue Form Sie nicht kennen, sondern nur wissen, dass er die Schlüssel $1, 2, 3, \ldots, 100$ enthält. Können Sie trotzdem wissen, welcher Schlüssel anschließend in der Wurzel des Baumes zu finden ist, wenn nun der Schlüssel 50 gelöscht wird? Wenn ja, dann tragen sie den Schlüssel hier ein, ansonsten schreiben Sie „nein“: [ ] Was wäre die Antwort, wenn wir 1 anstatt 50 löschen? [ ]

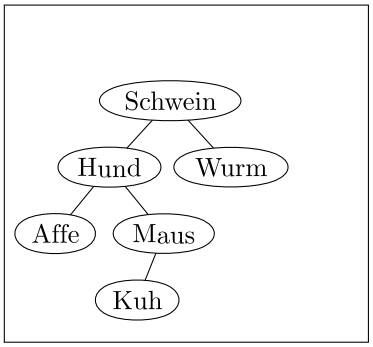
Begründen Sie Ihre Antworten nachvollziehbar:

Lösungsvorschlag









d) Wir wissen es. Das Löschen von 50 geht so vonstatten: Erst wird *splay*(50) ausgeführt. Danach wird eine splay-Operation auf dem größten Schlüssel im linken Unterbaum durchgeführt, also auf **49**, wodurch diese in die Wurzel wandert und danach auch dort bleibt. Wenn wir die 1 löschen, splayen wir zuerst die 1 in die Wurzel, können jedoch kein kleineres Kind anschließend in die Wurzel splayen. Die neue Wurzel wird dadurch das rechte Kind der 1, über welches wir keine Aussage treffen können.

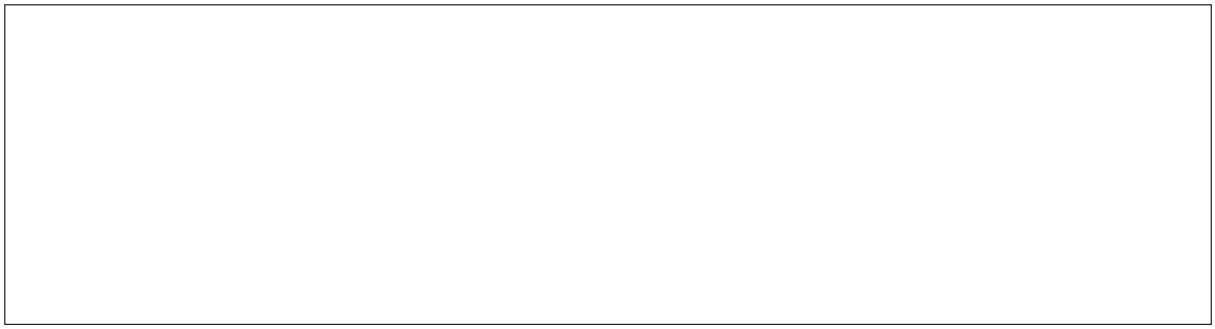
Aufgabe K4 (3+3+8 Punkte)

Gegeben ist ein ungerichteter Graph G mit n Knoten und m Kanten. Er ist durch Adjazenzlisten repräsentiert. Wir nennen einen ungerichteten Graphen *zusammenhängend*, wenn alle Knoten paarweise durch einen Pfad verbunden sind.

a) Wie kann man in linearer Zeit, also in $O(n + m)$ Schritten, feststellen, ob G zusammenhängend ist?

b) Wir nennen einen Graphen G *dreifach zusammenhängend,* wenn G zusammenhängend bleibt, selbst wenn man zwei beliebige Kanten entfernt. Zeichnen Sie zwei Graphen: links

einen Graphen, der zusammenhängend, aber nicht dreifach zusammenhängend ist und rechts einen Graphen, der dreifach zusammenhängend ist.



c) *Wir empfehlen, diese Teilaufgabe erst dann zu bearbeiten, wenn Sie bereits mit der Bearbeitung der restlichen Klausur fertig sind.*

Den dreifachen Zusammenhang eines Graphen kann man feststellen, indem man für alle $O(m^2)$ Kantenpaare testet, ob der Graph einfach zusammenhängend ist, nachdem man das Kantenpaar entfernt. Die Laufzeit dieses einfachen Verfahrens ist $O(m^2(n+m))$.

Entwerfen Sie ein Verfahren, welches dieselbe Aufgabe in $O(n^2(n+m))$ Schritten löst. Beschreiben Sie es ausreichend ausführlich und begründen Sie kurz seine Korrektheit und Laufzeit.

Lösungsvorschlag

a) Wir machen von einem beliebigen Startknoten in Zeit $O(n+m)$ eine Tiefensuche. Der Graph ist genau dann zusammenhängend, wenn jeder Knoten besucht wurde.

b) Links: ein Pfad mit zwei Knoten

Rechts: ein vollständig verbundender Graph mit vier Knoten.

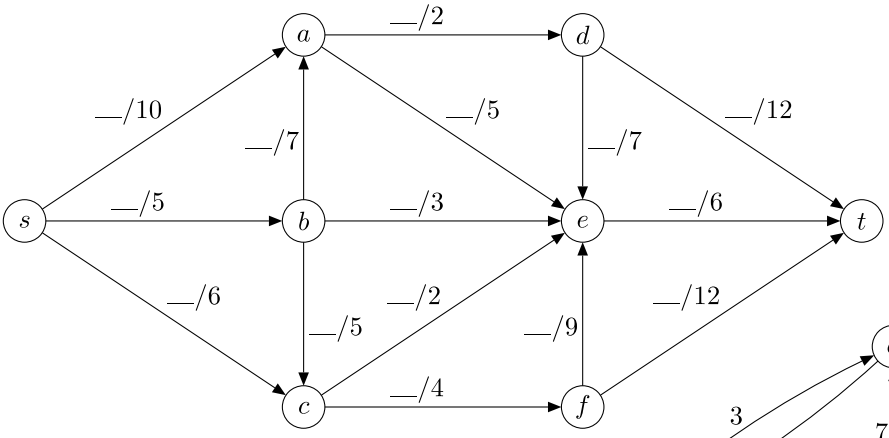
c) Für jedes Paar an Knoten s, t in G definieren wir ein Flussnetzwerk mit Quelle s und Senke t, indem wir jede ungerichtete Kante durch eine Hin- und Rückkante jeweils mit Kapazität eins ersetzen. Man sieht leicht, dass ein Graph zusammenhängend ist, genau dann wenn für jedes Knotenpaar s, t jeder Schnitt in dem entsprechenden Flussnetzwerk mindestens Kapazität eins hat.

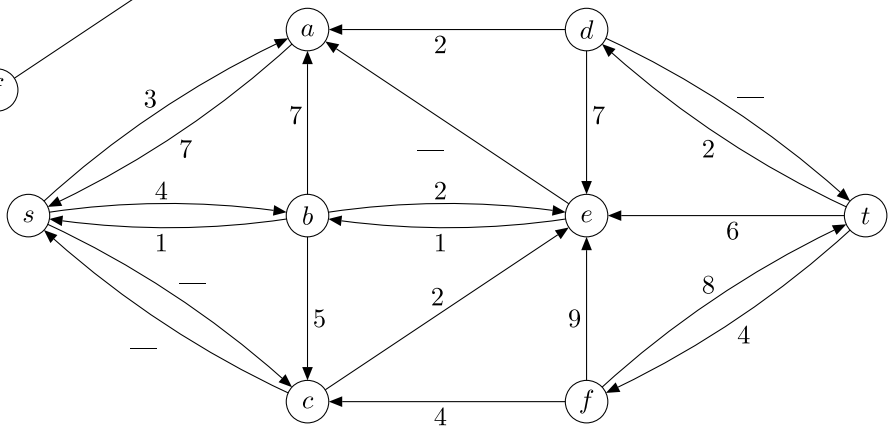
Wir gehen einen Schritt weiter, und zeigen, dass ein Graph dreifach zusammenhängend ist, genau dann wenn für jedes Knotenpaar s, t jeder Schnitt in dem entsprechenden Flussnetzwerk mindestens Kapazität drei hat.

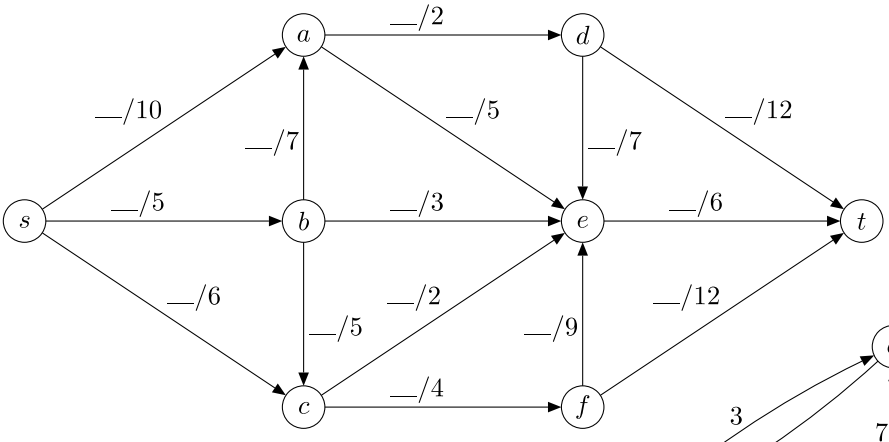
Angenommen für jedes Knotenpaar s, t hat jeder Schnitt in dem entsprechenden Flussnetzwerk mindestens Kapazität drei. Wenn wir zwei beliebige Kanten entfernen, dann hat jeder Schnitt immernoch Kapazität mindestens eins. Somit ist der Graph immernoch zusammenhängend. Für die Rückrichtung nehmen wir an, dass es ein Knotenpaar s, t mit einem s, t-Schnitt mit Kapazität null, eins, oder zwei gibt. Löschen wir die Kanten auf dem Schnitt, so ist der Graph nicht mehr zusammenhängend. Da wir maximal zwei Kanten gelöscht haben, ist der Graph nicht dreifach zusammenhängend.

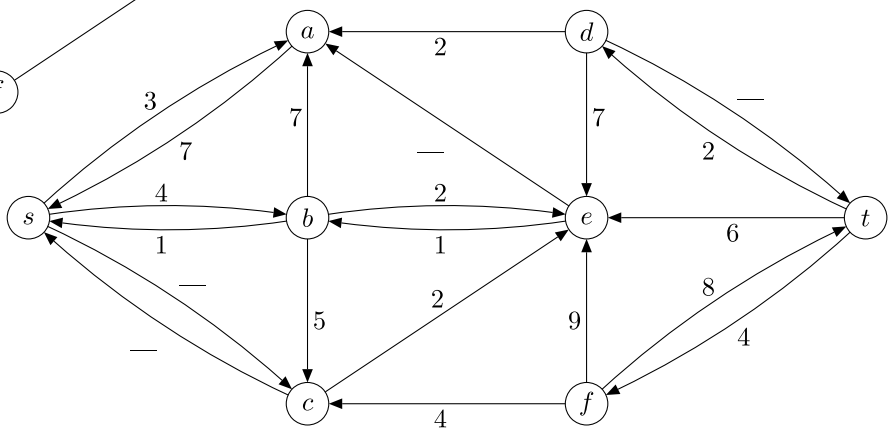
Es folgt, dass ein Graph dreifach zusammenhängend ist, genau dann wenn für jedes Knotenpaar s, t es einen s, t-Fluss mit Wert mindestens drei gibt. Dies können wir leicht algorithmisch testen: Wir enumerieren alle Knotenpaare s, t in Zeit $O(n^2)$. Danach versuchen wir drei Augmentierungen in dem entsprechenden Flussnetzwerk zu machen. Jede Augmentierung dauert Zeit $O(n+m)$.

Sie können folgende Grafiken als Schmierpapier verwenden, um Aufgabe 2 zu lösen. Bitte beachten Sie, dass eingezeichnete Lösungswege in diesen Grafiken nicht zur Bewertung herangezogen werden. Sollten Sie ihr Endergebnis dennoch in diese Grafiken eintragen wollen, so kennzeichnen Sie dies deutlich!









Datenstrukturen und Algorithmen, SS 2019
Prof. Dr. P. Rossmanith
S. Dollase, J. Dreier, H. Lotze

RWTH AACHEN UNIVERSITY

Datum: 06.09.2019

Klausur mit Lösungen 02

Datenstrukturen und Algorithmen, SS 2019
Prof. Dr. P. Rossmanith
S. Dollase, J. Dreier, H. Lotze

RWTH AACHEN UNIVERSITY

Datum: 06.09.2019

Klausur mit Lösungen 02

Datenstrukturen und Algorithmen, SS 2019
Prof. Dr. P. Rossmanith
S. Dollase, J. Dreier, H. Lotze

RWTH AACHEN UNIVERSITY

Datum: 06.09.2019

Klausur mit Lösungen 02

Datenstrukturen und Algorithmen, SS 2019
Prof. Dr. P. Rossmanith
S. Dollase, J. Dreier, H. Lotze

RWTH AACHEN UNIVERSITY

Datum: 06.09.2019

Klausur mit Lösungen 02

Datenstrukturen und Algorithmen, SS 2019
Prof. Dr. P. Rossmanith
S. Dollase, J. Dreier, H. Lotze

RWTH AACHEN UNIVERSITY

Datum: 06.09.2019

Klausur mit Lösungen 02