

Sommersemester 2023

Termin: 26. Juni 2022 Bearbeitungszeit: 120 min Hilfsmittel: keine	Bitte in Druckschrift ausfüllen:
	Name: _____
	Vorname: _____
	Matrikelnummer: _____
	Studiengruppe: <input type="radio"/> K <input type="radio"/> JÜL

- Füllen Sie dieses Deckblatt bitte vollständig aus und versehen Sie jedes Blatt mindestens mit Ihrer Matrikelnummer.
- Bitte beantworten Sie die Aufgaben auf den Aufgabenblättern (ggf. Rückseite verwenden).
- **Achtung!** Aufgaben 8 und 9 sind **Wahlaufgaben**. Sie müssen nur eine der beiden Varianten für das Erreichen einer sehr guten Note bearbeiten. Dasselbe gilt für die Aufgaben 6c und 6d, mit denen Sie Zusatzpunkte erwirtschaften können.
- Sie müssen die Aufgaben nicht in der vorgegebenen Reihenfolge abarbeiten! Am besten gehen Sie wie folgt vor:
 - verschaffen Sie sich einen Überblick über alle Aufgaben;
 - klassifizieren Sie die Aufgaben nach Schwierigkeitsgrad;
 - beginnen Sie mit den Aufgaben, die Ihnen am einfachsten erscheinen.
- Schreiben Sie bitte lesbar und verständlich mit einem dokumentenechten Stift (bitte **kein Bleistift / keine roten Stifte**).
- Geben Sie am Ende der Klausur auch sämtliche Hilfsblätter ab (bitte ebenfalls mit Matrikelnummer versehen).
- Diese Klausur besteht aus **19 Seiten** (incl. Deckblatt)

Viel Erfolg!

Bewertung											
Aufgabe	1	2	3	4	5	6	7	8	9	Gesamt	Note
Max. Punkte	10	8	10	8	10	9+5	15	30		100+5	
Erreichte Punkte											
Unterschrift Prüfer											

Name:

Matr.-Nr:

1. Aufgabe

10 Punkte

Notieren Sie die Antworten in ein oder mehreren Stichworten oder einem kurzen Satz:

(a) Kreuzen Sie an: welche der nachfolgenden regulären Ausdrücke erkennen aa?

Ausdruck	erkennt aa
$(a b)^+c?aa$	<input type="radio"/>
$a*b*c?aa$	<input type="radio"/>
$ab*c?a?$	<input type="radio"/>

(b) Ist Quick-Sort ein stabiler Sortieralgorithmus?

(c) Ordnen Sie den folgenden Code-Schnipseln einen Sortier-Algorithmus zu:

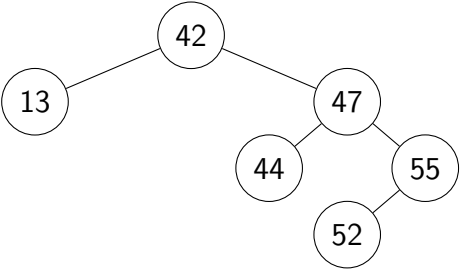
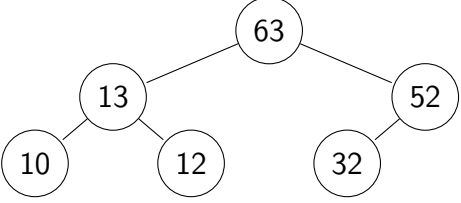
Code-Schnipsel	Sortier-Algorithmus
<pre>int c = magic(l, r, f); sort(l, c-1, f); sort(c+1, r, f);</pre>	
<pre>int middle = (l+r)/2; sort(l, middle, feld); sort(middle+1, r, feld); magic(l, middle, r, feld);</pre>	

(d) Ist die Adjazenzmatrix von ungerichteten Graphen symmetrisch?

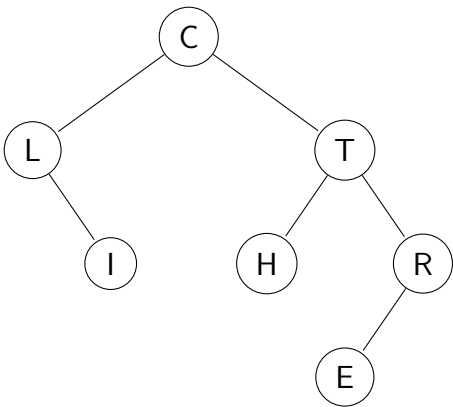
(e) Wie viele Rotationen werden beim Löschen in einem AVL-Baum der Höhe h maximal durchgeführt?

(f) Auf welche Graphen dürfen Sie den Floyd-Warshall-Algorithmus **nicht** anwenden?

(g) Kreuzen Sie in der nachfolgenden Tabelle an, welche Eigenschaften die jeweiligen Bäume erfüllen.

Baum	Heap	AVL-Baum	Binärer Suchbaum
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

(h) Welche Durchlauf-Strategie erzeugt für den nachfolgenden Baum die Ausgabe LICHTER?



Name:	Matr.-Nr:
-------	-----------

2. Aufgabe

8 Punkte

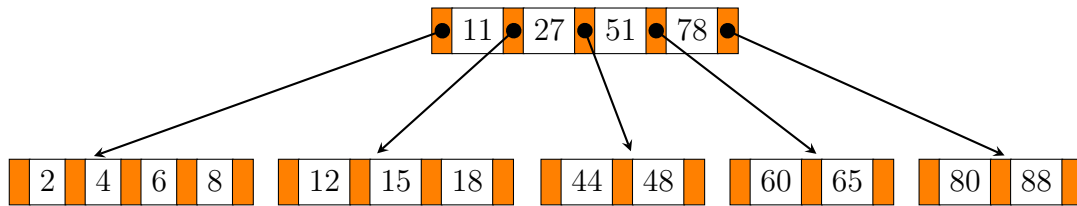
Sortieren Sie das folgende Feld mit *Quick-Sort*. Dabei sollte als Pivot-Element immer das am weitesten rechts stehende Element gewählt werden. Schreiben Sie nach jeder Vertauschung das Zwischenergebnis auf und markieren Sie die Teilfolgen, für die rekursive Aufrufe durchgeführt werden.

18	2	72	7	8	57	9	10	12
----	---	----	---	---	----	---	----	----

3. Aufgabe

10 Punkte

Betrachten Sie den folgenden B-Baum der Ordnung 2:



Wichtig! Aufgabenteile a-e verwenden als Ausgangspunkt immer diesen Baum!

(a) Entfernen Sie aus dem B-Baum den Wert 27 und zeichnen Sie den entstehenden B-Baum.

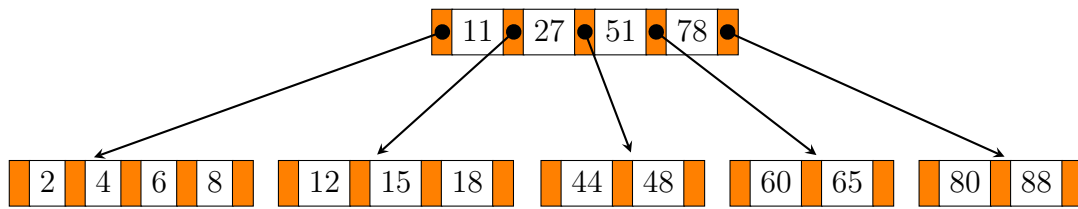
(b) Entfernen Sie aus dem **ursprünglichen** B-Baum den Wert 44 und zeichnen Sie den entstehenden B-Baum.

Name:

Matr.-Nr:

Fortsetzung von Aufgabe 3.

Betrachten Sie den folgenden B-Baum der Ordnung 2:



Wichtig! Aufgabenteile a-e verwenden als Ausgangspunkt immer diesen Baum!

(c) Fügen sie in den **ursprünglichen** Baum den Wert 81 ein.

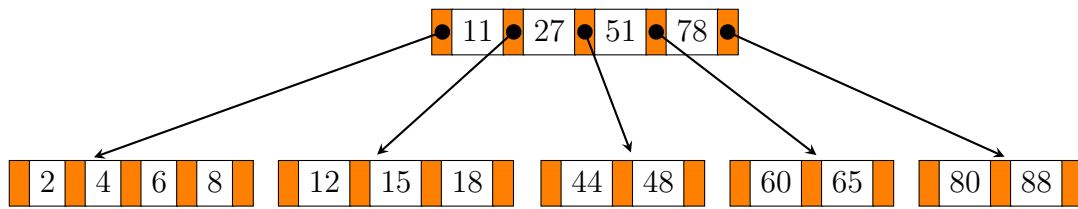
(d) Entfernen Sie aus dem **ursprünglichen** B-Baum den Wert 65 und zeichnen Sie den entstehenden B-Baum.

Name:

Matr.-Nr:

Fortsetzung von Aufgabe 3.

Betrachten Sie den folgenden B-Baum der Ordnung 2:



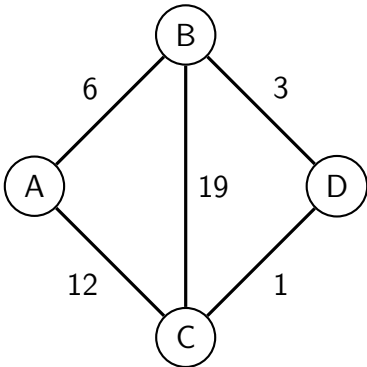
Wichtig! Aufgabenteile a-e verwenden als Ausgangspunkt immer diesen Baum!

(e) Fügen Sie in den **ursprünglichen** B-Baum den Wert 10 ein und zeichnen Sie den entstehenden B-Baum.

5. Aufgabe

10 Punkte

Betrachten Sie den folgenden Graphen G :



Berechnen Sie mit Hilfe des Floyd-Warshall-Algorithmus die Längen aller kürzesten Pfade in G . Geben Sie hierzu die Werte der Distanz-Matrix und der Vorgänger-Matrix Π jeweils nach jedem Durchlauf der Hauptschleife an (trennen Sie Pfadlänge und Vorgängerknoten durch einen Schrägstrich):

0	A	B	C	D
A	∞ nil	6 A	12 A	∞ nil
B	6 B	∞ nil	19 B	3 B
C	12 C	19 C	∞ nil	1 C
D	∞ nil	3 D	1 D	∞ nil

2	A	B	C	D
A				
B				
C				
D				
4	A	B	C	D

A				
B				
C				
D				

1	A	B	C	D
A				
B				
C				
D				
3	A	B	C	D

A				
B				
C				
D				
5	A	B	C	D

A				
B				
C				
D				

6. Aufgabe

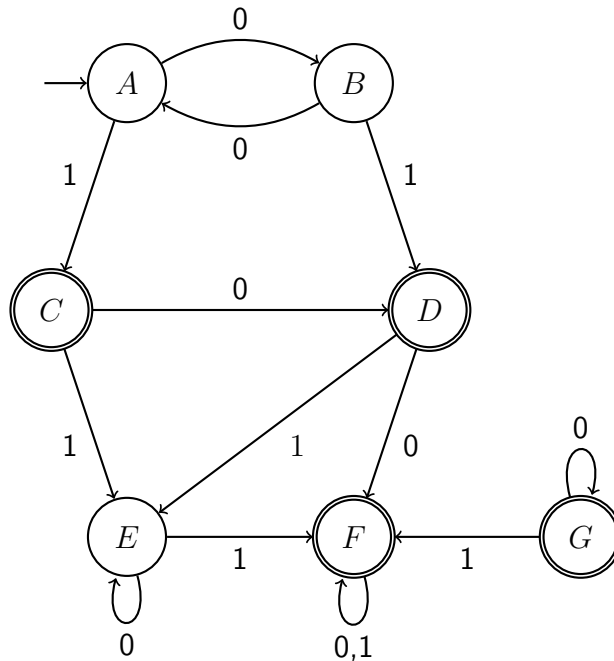
9+5 Punkte

- (a) Zeichnen Sie einen deterministischen endlichen Automaten, der die Sprache

$$L_1 = \{w \in \Sigma_{Bool}^* \mid w = u001001v \text{ mit } u, v \in \Sigma_{Bool}^*\}$$

erkennt.

- (b) Minimieren Sie den folgenden endlichen Automaten mit einem in der Vorlesung vorgestellten Verfahren.
Wichtig! Andere Verfahren werden nicht akzeptiert!



Bei den folgenden Aufgaben handelt es sich um **Zusatzaufgaben**, von denen Sie **eine** auswählen können. Sollten Sie beide beantworten, erhalten Sie **keine** Punkte.

- (c) Sei $\Sigma = \{a, b, c, t, x\}$ ein Alphabet. Wie viele Wörter enthält die Sprache

$$L_2 = \{w \in \Sigma^* \mid |w| = 7 \wedge w \text{ enthält } \textit{atta} \text{ als Teilwort}\}$$

- (d) Geben Sie eine induktive Funktionsdefinition für eine Funktion $\text{delChar} : \Sigma^* \times \Sigma \rightarrow \Sigma^*$ an, welche alle Vorkommen eines Symbols aus einem Wort entfernt.

Name:	Matr.-Nr:
-------	-----------

Name:	Matr.-Nr:
-------	-----------

7. Aufgabe

15 Punkte

Gegeben ist eine Klasse für eine doppelt verkettete Liste. Die Attribute der Klasse sind:

```
private Node first;  
private Node last;  
private int elementCnt;
```

Dabei gibt `elementCnt` die Anzahl der Elemente an, die in der Liste vorhanden sein sollen. Die Klasse `Node` wurde folgendermaßen implementiert:

```
public class Node {  
    public int val;  
    public Node next;  
    public Node prev;  
}
```

Ergänzen Sie die Listenklasse um eine Methode

```
public boolean isValid()
```

die prüft, ob es sich um eine korrekt verkettete Liste mit `elementCnt` Elementen handelt.

Dabei sollen folgende Eigenschaften überprüft werden:

- Vorwärts- und Rückwärtsverkettungen entsprechen einander.
- Es gibt keine Schleifen in der Liste. Die komplette Liste darf jedoch zu einem Ring geschlossen sein.

Name:	Matr.-Nr:
-------	-----------

Name:

Matr.-Nr:

8. Aufgabe (Variante A)

15 Punkte

Gegeben sei die Rekursionsgleichung

$$T(n) = \begin{cases} 0 & \text{falls } n = 0 \\ 1 & \text{falls } n = 1 \\ 3 \cdot T(n-1) - 2 \cdot T(n-2) & \text{sonst} \end{cases}$$

Benutzen Sie die Technik der Erzeugenden Funktion um eine geschlossene Form für $T(n)$ zu finden.

Name:	Matr.-Nr:
-------	-----------

Name:	Matr.-Nr:
-------	-----------

9. Aufgabe (Variante A)

15 Punkte

Gegeben sei ein binärer Suchbaum mit ganzzahligen Schlüsseln. Reichern Sie die Datenstruktur so an, dass Sie den Mittelwert der Schlüssel in einem beliebigen Unterbaum in konstanter Zeit berechnen können.

Welche zusätzlichen Informationen müssen an den Knoten des Suchbaums gespeichert werden und was muss beim Einfügen und Löschen von Knoten beachtet werden, damit die Information immer korrekt ist? Beschreiben Sie die Löscho- und Einfügeoperation informell.

Name:	Matr.-Nr:
-------	-----------

8. Aufgabe (Variante B)

15 Punkte

Schreiben Sie eine Methode

```
public boolean subsetIsSum(int[] set, int sum)
```

die überprüft, ob es eine Teilmenge von `set` gibt, die aufsummiert den Wert `sum` ergibt. Gehen Sie davon aus, dass `sum` immer größer als 0 ist.

Hinweis: Sie müssen alle möglichen Teilmengen durchprobieren. Das gesamte Feld ist ebenfalls eine Teilmenge von `set`.

9. Aufgabe (Variante B)

15 Punkte

Gegeben seien folgende Klassendefinitionen für einen binären Baum:

```
public class Node {
    public Node left, right;
    public double value;

    public Node (double wert) {
        value = wert;
        left = null;
        right = null;
    }
}
```

```
public class Tree {
    private Node root;

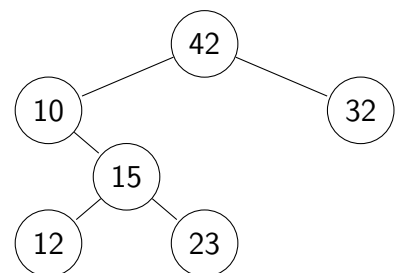
    public Tree (Node node) {
        root = node;
    }
}
```

Fügen Sie der Klasse Tree eine Methode

```
public String getMaximumPath()
```

hinzu, die den Pfad (Wurzel→Blatt) zurückgibt, bei dem die Summe der Elemente maximal ist. Der Pfad soll folgendermaßen codiert sein: „r“ steht für „gehe zum rechten Kind“, „l“ steht für „gehe zum linken Kind“.

Beispiel: Der maximale Pfad im Baum rechts ist 42-10-15-23. Man geht von der Wurzel aus nach links, nach rechts und wieder nach rechts. Daher ist das Ergebnis: „lrr“.



Name:	Matr.-Nr:
-------	-----------