

✓ RS  

---

**Prüfungsteilnehmer**

**Prüfungstermin**

**Einzelprüfungsnummer**

---

**Kennzahl:** \_\_\_\_\_

**Kennwort:** \_\_\_\_\_

**Arbeitsplatz-Nr.:** \_\_\_\_\_

**Frühjahr  
2013**

**46115**

---

**Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen  
— Prüfungsaufgaben —**

---

**Fach:** Informatik (Unterrichtsfach)

**Einzelprüfung:** Th. Informatik, Algorith./Datenstr.

**Anzahl der gestellten Themen (Aufgaben):** 2

**Anzahl der Druckseiten dieser Vorlage:** 7

---

**Bitte wenden!**

**Thema Nr. 1****Aufgabe 1:****Reguläre Sprachen**

Sei  $L$  die Menge aller Worte der Länge  $\geq 2$  über dem Alphabet  $\{a,b\}$ , bei denen das zweite und das zweitletzte Zeichen identisch sind.

- a) Geben Sie alle Worte der Länge  $\leq 4$  von  $L$  an.
- b) Zeigen Sie, dass  $L$  regulär ist.

**Aufgabe 2:****Pumping Lemma**

- a) Formulieren Sie das Pumping Lemma für reguläre Sprachen.
- b) Zeigen Sie mit Hilfe des Pumping Lemmas, dass die Sprache  $\{a^{i+j} b^i c^j \mid i, j \geq 0\}$  nicht regulär ist.

**Aufgabe 3:****Kontextfreie Sprachen**

Zeigen Sie, dass die Sprache  $L = \{a^i b^p c^q d^j \mid i, j, p, q \geq 0 \text{ und } i=j \text{ und } p \geq q\}$  kontextfrei ist.

Wenn Sie eine passende Grammatik  $G$  / einen passenden Automaten  $M$  angegeben haben, brauchen Sie nicht zu beweisen, dass  $L=L(G)$  bzw.  $L=L(M)$  ist.

**Aufgabe 4:****Turingmaschinen**

Sei  $L = \{u v \mid u \in \{a,b\}^*, v \in \{c,d\}^*, \#_a(u) = \#_c(v) \text{ und } \#_b(u) = \#_d(v)\}$   
wobei  $\#_a(u)$  die Anzahl der in  $u$  vorkommenden  $a$ 's ist.

- a) Geben Sie eine Turingmaschine  $M$  an, die  $L$  erkennt.  
Beschreiben Sie in Worten, wie Ihre Turingmaschine arbeitet.
- b) Welche Laufzeit (Zeitkomplexität) hat Ihre Turingmaschine (in  $O$ -Notation).  
Begründen Sie Ihre Angabe auf der Grundlage der Beschreibung.

**Aufgabe 5:****O-Notation**

Gegeben seien die Funktionen  $f: \mathbb{N} \rightarrow \mathbb{N}$  und  $g: \mathbb{N} \rightarrow \mathbb{N}$ , wobei  $f(n)=2n^2+2$  und  $g(n)=3n(n^2-1)$ .  
Geben Sie an, welche der folgenden Aussagen gelten. Beweisen Sie Ihre Angaben.

- a)  $f(n) \in O(g(n))$
- b)  $g(n) \in O(f(n))$

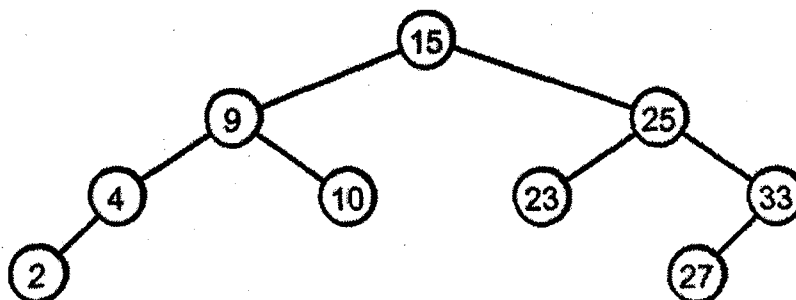
**Fortsetzung nächste Seite!**

**Aufgabe 6:****Sortieren**

- Erläutern Sie, wie Bubblesort arbeitet.
- Sortieren Sie die folgende Liste von Zahlen mit Bubblesort und geben Sie die Zwischenergebnisse nach jeder Runde an.  
2,10,3,8,1,9,5,4,6,7
- Für welche Eingaben ist die Laufzeit von Bubblesort maximal (der worst case)?

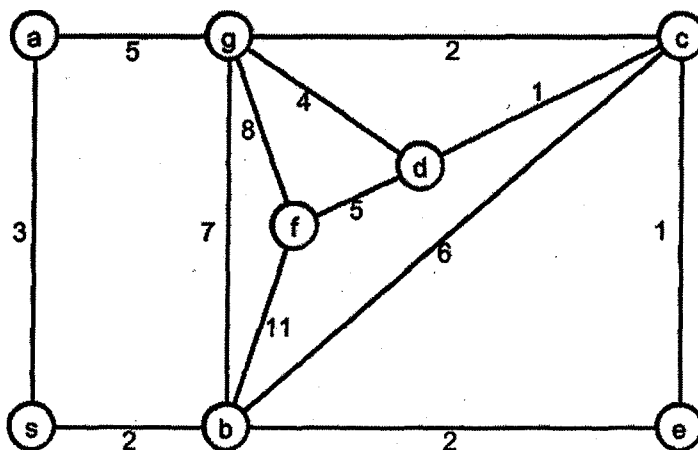
**Aufgabe 7:****AVL-Bäume**

Gegeben sei der folgende AVL-Baum T. Führen Sie auf T folgende Operationen durch: Fügen Sie zunächst den Wert 1 in T ein. Fügen Sie anschließend den Wert 26 in T ein. Balancieren Sie nach jeder Operation falls nötig und geben Sie den entstandenen Baum (als Zeichnung) an.

**Aufgabe 8:****Minimaler Spannbaum**

Gegeben sei der unten stehende ungerichtete Graph  $G = (V, E)$  mit positiven Kantenlängen  $l(e)$  für jede Kante  $e \in E$ .

Berechnen Sie mit einem Algorithmus Ihrer Wahl einen minimalen Spannbaum von G. Verwenden Sie den Knoten s als Startknoten.



## Thema Nr. 2

## Aufgabe 1:

Sei  $\Sigma = \{0,1,\$ \}$  und sei  $\mathbb{N}_0 = \{0,1,2,\dots\}$ .

a) Sei

$$L_1 = \{0^n \$ 1^{2n} \mid n \in \mathbb{N}_0\} \cup \{0^{3n} \$ 1^n \mid n \in \mathbb{N}_0\}.$$

Beispiele:  $00\$1111 \in L_1$ ,  $\$ \in L_1$ ,  $000\$1 \in L_1$ .

(a1) Zeigen Sie, dass  $L_1$  kontextfrei ist, indem Sie eine kontextfreie Grammatik  $G$  angeben mit  $L(G) = L_1$ , und

(a2) begründen Sie, warum Ihre Grammatik *genau* die Sprache  $L_1$  erzeugt.

b) Formulieren Sie das Pumping-Lemma für kontextfreie Sprachen:

*„Sei  $L$  eine kontextfreie Sprache über dem Alphabet  $\Sigma$ . Dann gibt es ...“*

c) Zeigen Sie, dass die Sprache  $L_1$  die Eigenschaften des Pumping-Lemmas für kontextfreie Sprachen erfüllt.

## Aufgabe 2:

a) Geben Sie einen deterministischen endlichen Automaten an (Zeichnung), der die Sprache

$$L_2 = \{a^i \mid i \in \mathbb{N}, i \text{ ist durch } 3, \text{ aber nicht durch } 2 \text{ teilbar}\}$$

akzeptiert. ( $\mathbb{N} = \{1,2,\dots\}$ )

b) Beweisen oder widerlegen Sie die folgende Behauptung:

Beh.: Ist die Sprache  $L$  regulär, dann ist auch jede echte Teilmenge von  $L$  regulär.

c) Zu einem Wort  $w = a_1 a_2 \dots a_n$  sei

$$\text{spiegelung}(w) = \{a_n \dots a_1\}.$$

Außerdem sei  $\text{spiegelung}(\varepsilon) = \{\varepsilon\}$ .

Beweisen oder widerlegen Sie die folgende Behauptung:

Beh.: Ist die Sprache  $L$  regulär, dann ist auch

$$\text{spiegelung}(L) = \bigcup_{w \in L} \text{spiegelung}(w)$$

regulär.

**Fortsetzung nächste Seite!**

**Aufgabe 3: „Binärer Suchbaum“**

Ein *Suchbaum* ist ein binärer Baum mit der Eigenschaft, dass für jeden Knoten  $n$  alle Datenelemente im linken Teilbaum kleiner und alle Elemente im rechten Teilbaum größer sind als das Element des Knotens  $n$ .

- Fügen Sie in einen anfangs leeren binären Suchbaum der Reihe nach die Datenelemente 7, 0, 9, 11, 18, 4, 5, 3, 13, 24, 2 ein. Zeichnen Sie nur den Endzustand.
- Geben Sie eine Einfügereihenfolge (keinen Baum) für obige Zahlen an, bei der die Höhe *minimal* wird.
- Geben Sie eine weitere Einfügereihenfolge dieser Datenelemente an, bei der die Höhe des entstehenden Baums *maximal* wird.
- Fügen Sie die Zahlen aus Teilaufgabe a) in der gegebenen Reihenfolge in einen anfangs leeren AVL-Baum ein. Zeichnen Sie dazu nur den jeweils entstandenen Baum zu folgenden „Zeitpunkten“:
  - Unmittelbar vor dem Einfügen der Zahl 18
  - Nach dem Einfügen der Zahl 18 jeweils vor und nach der Rotation
  - Nach dem Einfügen aller Zahlen (also den vollständigen AVL-Baum)
- Fügen Sie nun Ihrem vollständigen AVL-Baum aus der letzten Teilaufgabe noch die Zahl 30 hinzu und geben Sie den schließlich entstandenen AVL-Baum an.

**Aufgabe 4: „Streuspeicherung“**

Die Werte 7, 0, 9, 11, 18, 4, 5, 3, 13, 24, 2 sollen in eine Hashtabelle der Größe 11 (Fächer 0 bis 10) eingetragen werden. Die zur Hashfunktion  $h(x) = (7 \cdot x) \% 11$  gehörenden Schlüssel sind in der folgenden Tabelle bereits ausgerechnet:

x	7	0	9	11	18	4	5	3	13	24	2
h(x)	5	0	8	0	5	6	2	10	3	3	3

- Fügen Sie die oben genannten Schlüssel in der vorgegebenen Reihenfolge in einen Streuspeicher ein, welcher zur Kollisionsauflösung verkettete Listen verwendet, und stellen Sie die endgültige Streutabelle dar.
- Fügen Sie die gleichen Schlüssel mit linearem Sondieren bei Schrittweite +1 zur Kollisionsauflösung in eine neue Hash-Tabelle ein. Geben Sie für jeden Schlüssel an, auf welche Felder beim Einfügen zugegriffen wird und ob Kollisionen auftreten. Geben Sie die gefüllte Streutabelle an.
- Wie hoch ist der „Load“-Faktor (die Belegung) der Hashtabelle aus a) bzw. b) in Prozent? Können Sie weitere Schlüssel einfügen?
- Würden Sie sich bei dieser Zahlensequenz für das Hashing-Verfahren nach a) oder nach b) entscheiden? Begründen Sie kurz Ihre Entscheidung.

**Fortsetzung nächste Seite!**

**Aufgabe 5: „Graphen“**

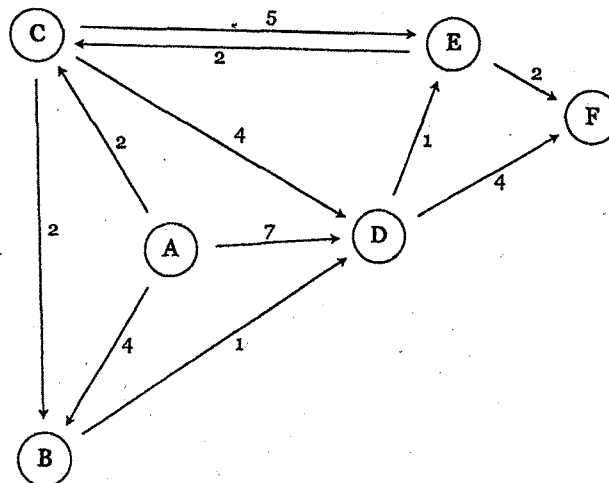
Gegeben seien folgende ungerichtete Graphen in textueller Notation, wobei die erste Menge die Menge der Knoten und die zweite Menge die Menge der Kanten ist:

$$G_1 = ( \{1,2,3,4,5,6\}, \{[1,2], [1,4], [2,3], [3,4], [4,5], [4,6], [5,6]\} )$$

$$G_2 = ( \{1,2,3,4,5,6\}, \{[1,2],[1,3],[2,3],[4,5],[4,6],[5,6]\} )$$

$$G_3 = ( \{1,2,3,4,5,6\}, \{[1,2],[1,6],[2,3],[2,5],[3,4],[4,5],[5,6]\} )$$

- Zeichnen Sie die obigen Graphen.
- Erstellen Sie zu jedem Graphen die zugehörige Adjazenzmatrix mit X als Symbol für eine eingetragene Kante.
- Für welche(n) der drei Graphen existiert ein Eulerzyklus? Begründen Sie kurz, wieso es bei dem bzw. den anderen Graphen keinen Eulerzyklus gibt.
- Betrachten Sie nun folgenden gerichteten Graphen  $G_4$ :



Bestimmen Sie die kürzeste Entfernung vom Knoten A zu jedem anderen Knoten des Graphen. Verwenden Sie dazu den Algorithmus von Dijkstra und tragen Sie Ihre einzelnen Rechenschritte in eine Tabelle folgender Form ein (schreiben Sie neben jede Zeile die Prioritätswarteschlange der noch zu bearbeitenden Knoten, priorisiert nach ihren Wegkosten):

A	B	C	D	E	F	Warteschlange
0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	A
...						

- Nennen Sie einen geeigneten Algorithmus, um einen minimalen Spannbaum für den obigen Graphen  $G_4$  zu ermitteln und wenden Sie diesen schrittweise (z.B. tabellarisch) an.

**Fortsetzung nächste Seite!**

**Aufgabe 6: Sortierverfahren**

- a) Vervollständigen Sie die folgende Sortierung mit MergeSort (Sortieren durch Mischen) – beginnen Sie dabei Ihren „rekursiven Abstieg“ immer im linken Teilfeld:

**D** | 40    5    89    95    85    84 || 14    25    20    52    7    71 |

Notation: Markieren Sie Zeilen mit **D**(ivide), in denen das Array zerlegt wird, und mit **M**(erge), in denen Teilarrays zusammengeführt werden. Beispiel:

<b>D</b>		82		89	44	
<b>D</b>		82		89		44
<b>M</b>		82		44	89	
<b>M</b>		44	82	89		

- b) Sortieren Sie mittels HeapSort (Haldensortierung) die folgende Liste weiter:

<b>I</b>		99	63	91	4	36	81	76	
<b>R</b>		76	...					99	

Notation: Markieren Sie die Zeilen wie folgt:

**I**: Initiale Heap-Eigenschaft hergestellt (größtes Element am Anfang der Liste).

**R**: Erstes und letztes Element getauscht und letztes „gedanklich entfernt“.

**S**: Erstes Element nach unten „versickert“ (Heap-Eigenschaft wiederhergestellt).