

---

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
--------------------	----------------	----------------------

---

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

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

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

**Herbst  
2009**

**66115**

---

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

---

Fach: **Informatik(vertieft studiert)**

Einzelprüfung: **Theoretische Informatik, Algorithmen**

Anzahl der gestellten Themen (Aufgaben): 2

Anzahl der Druckseiten dieser Vorlage: 7

---

Bitte wenden!

Thema Nr. 1  
(Aufgabengruppe)

Es sind alle Aufgaben dieser Aufgabengruppe zu bearbeiten!

**Aufgabe 1:**

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

a) Sei

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

Beispiele:  $00\$111111 \in L_1$ ,  $\$ \in L_1$ ,  $000\$1 \in L_1$ .

- i) Zeigen Sie, dass  $L_1$  kontextfrei ist, indem Sie eine kontextfreie Grammatik  $G$  angeben mit  $L(G) = L_1$ .
  - ii) 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 ...“
- c) Zeigen Sie z. B. mittels Pumping-Lemma für kontextfreie Sprachen, dass die Sprache

$$L_2 = \{0^n 1^{3n} \$0^n \mid n \in \mathbb{N}_0\}$$

nicht kontextfrei ist.

Beispiel:  $00\$111111\$00 \in L_2$

**Aufgabe 2:**

- a) Definieren Sie die zum Halteproblem für Turing-Maschinen gehörende Menge  $H$ .
- b) Gegeben sei das folgende Problem  $E$ :  
„Entscheide, ob es für die deterministische Turing-Maschine mit der Gödelnummer  $n$  eine Eingabe gibt, für die die Turing-Maschine hält.“  
Zeigen Sie, dass  $E$  nicht entscheidbar ist. Benutzen Sie, dass  $H$  aus (a) nicht entscheidbar ist.
- c) Angenommen, es wurde gezeigt, dass  $P = NP$  ist. Zeigen Sie, dass dann *jede* Sprache  $L \in P$  sogar NP-vollständig ist.

**Aufgabe 3:**

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

$$L_3 = \{a^i b^j \mid i, j \in \mathbb{N}_0, i \text{ und } j \text{ sind durch } 3 \text{ teilbar}\}$$

akzeptiert. ( $\mathbb{N}_0 = \{0, 1, 2, \dots\}$ )

Beispiele:  $aaabbbbb \in L_3$ ,  $aaabbb \in L_3$ ,  $\varepsilon \in L_3$ .

- b) Beweisen oder widerlegen Sie die folgende Behauptung:

Ist die Sprache  $L$  regulär, dann ist auch jede Obermenge von  $L$  regulär.

- c) Zu einem Wort  $w = a_1 a_2 a_3 \dots a_n$  ist

$$\text{suffix}(w) = \{\varepsilon, a_n, a_{n-1}a_n, a_{n-2}a_{n-1}a_n, \dots, a_1 a_2 a_3 \dots a_n\}$$

die Menge aller Suffixes von  $w$ .

Beweisen oder widerlegen Sie die folgende Behauptung:

Ist die Sprache  $L$  regulär, dann ist auch

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

regulär.

**Aufgabe 4:**

Erklären Sie die jeweils wesentliche(n) Idee(n) folgender Informatik-Konzepte in jeweils maximal drei Sätzen:

- a) Adjazenzliste
- b) Mergesort
- c) Lineare Suche
- d) Rekursive Datenstruktur

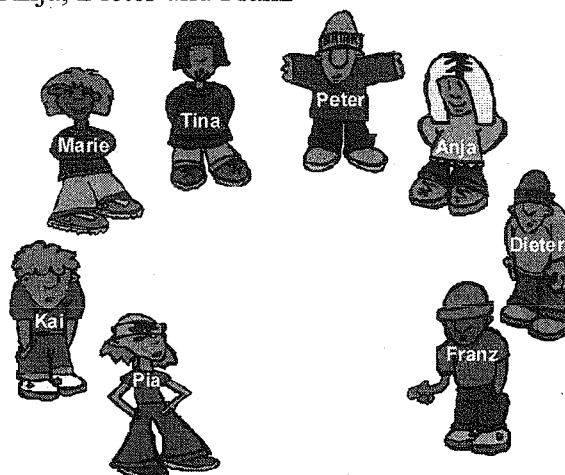
**Aufgabe 5:**

Gegeben seien  $n$  in einem Kreis stehende Kinder, die einen Abzählreim mit  $k$  Silben spielen. Dabei wird ausgehend von einem ersten Kind mit jeder weiteren Silbe des Abzählreims im Kreis herum (im Uhrzeigersinn) auf das jeweils nächste Kind gezeigt. Das Kind, auf das bei der letzten Silbe des Reims gezeigt wird, scheidet aus. Anschließend wird mit dessen Nachfolger fortgesetzt und der Abzählreim erneut gesprochen. Der Nachfolger eines Kindes sei dabei das Kind, das im Uhrzeigersinn hinter diesem steht unter der Annahme, dass alle in die Mitte des Kreises schauen. Das Ganze wird so lange durchgespielt, bis nur noch ein Kind übrig ist. Dieses hat gewonnen.

Gegeben sei der Abzählreim:

1, 2, 3, 4, 5, 6, 7  
eine alte Frau kocht Rüben,  
eine alte Frau kocht Speck -  
und du bist weg!

Im Kreis stehen dabei in dieser Reihenfolge folgende Kinder:  
Pia, Kai, Marie, Tina, Peter, Anja, Dieter und Franz



- a) Wer gewinnt, wenn bei Pia begonnen wird?
- b) Geben Sie in einer gängigen Programmiersprache oder als Pseudocode einen gut kommentierten Algorithmus an, der ausgehend von einer Menge von  $n$  Kindern, einem Kind, das startet, und einem Abzählreim mit  $k$  Silben das gewinnende Kind ermittelt. Ihre Lösung soll als Datenstrukturen nur einfache Variablen und Arrays verwenden, aber keine Zeigerstrukturen.
- c) Welche Laufzeit- und welche Platzkomplexität hat Ihr Algorithmus?

### Aufgabe 6: Datenstruktur für Intervalle

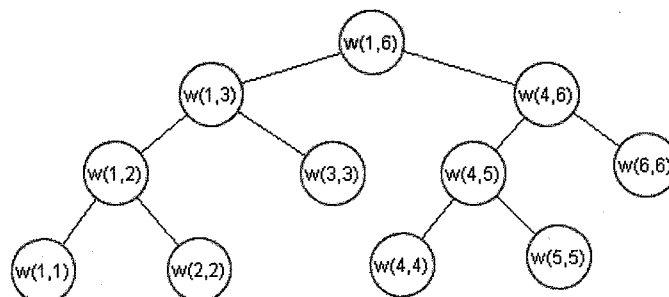
Sei  $M = \{1, 2, \dots, n\}$  eine Menge aufeinander folgender natürlicher Zahlen. Für jedes  $i \in M$  sei ein Wert  $w(i)$  definiert. Gesucht ist dann ein hinsichtlich Speicherbedarf und Laufzeit effizienter Algorithmus, der zu einem gegebenen Intervall  $[i, j]$  mit  $i, j \in M$  den Wert

$$w(i, j) = \begin{cases} w(i) + \dots + w(j) & , \text{ falls } i < j \\ 0 & , \text{ falls } i > j \\ w(i) & , \text{ falls } i = j \end{cases}$$

berechnet.

Der Algorithmus soll aus zwei Phasen bestehen:

1. Aufbau einer geeigneten Datenstruktur (Preprocessing).
  2. Berechnung eines Wertes  $w(i, j)$ .
- a) Formulieren Sie gut kommentiert in einer gängigen Programmiersprache oder als Pseudocode einen Algorithmus, der dieses Problem mit Hilfe eines eindimensionalen Arrays löst, das die Werte  $w(i)$  nach  $i$  aufsteigend sortiert speichert. Geben Sie mit Begründung die Laufzeit, den der Algorithmus nach dem Preprocessing in Phase 2 benötigt, und den allgemeinen Speicherplatzbedarf an.
  - b) Formulieren Sie nun gut kommentiert in einer gängigen Programmiersprache oder als Pseudocode einen Algorithmus, der dieses Problem mit Hilfe eines zweidimensionalen Arrays löst. Geben Sie mit Begründung die Laufzeit, die der Algorithmus nach dem Preprocessing in Phase 2 benötigt, und den allgemeinen Speicherplatzbedarf an.
  - c) Eine effiziente Möglichkeit zur Lösung dieses Problems ist die Verwendung eines so genannten Segmentbaumes. Für das Beispiel  $M = \{1, 2, 3, 4, 5, 6\}$  sieht ein solcher wie folgt aus:



Formulieren Sie wiederum gut kommentiert in einer gängigen Programmiersprache oder Pseudocode einen Algorithmus, der das Problem mit Hilfe eines solchen Segmentbaumes löst. Geben Sie dann mit Begründung die Laufzeit, die der Algorithmus nach dem Preprocessing in Phase 2 benötigt, und den allgemeinen Speicherplatzbedarf an.

Thema Nr. 2  
(Aufabengruppe)

Es sind alle Aufgaben dieser Aufabengruppe zu bearbeiten!

**Aufgabe 1:**

Stufen Sie die folgenden Sprachen über dem Alphabet  $\Sigma = \{a, b, c\}$  in die Chomsky-Hierarchie ein. Begründen Sie jeweils Ihre Antwort.

- a)  $L_1 = \{a^n b^m c^k \mid n, m, k \geq 1\}$
- b)  $L_2 = \{a^n b^m c^n \mid m, n \geq 1\}$
- c)  $L_3 = \{a^n b^n c^n \mid n \geq 1\}$
- d)  $L_4 = \{(ab)^n a (ba)^n \mid n \geq 1\}$
- e)  $L_5 = \{w \mid w \text{ codiert eine Turingmaschine, deren Ausführung bei leerer Eingabe hält}\}$
- f)  $L_6 = \{w \mid w \text{ codiert eine Turingmaschine, deren Ausführung bei leerer Eingabe nicht hält}\}$

Bei den letzten zwei Aufgabenteilen ist eine sinnvolle Codierung von Turingmaschinen durch Zeichenketten über  $\Sigma$  zugrunde gelegt.

**Aufgabe 2:**

- a) Es seien  $L_1$  und  $L_2$  reguläre Sprachen über  $\Sigma = \{a, b, c\}$ . Begründen Sie, dass die folgende Sprache dann auch regulär ist:

$$L = \{w \mid \text{für alle Zerlegungen } w = uvx \text{ gilt: } u \in L_1 \Rightarrow x \in L_2\}$$

Hinweis: Zeigen Sie zunächst, dass das Komplement von  $L$  regulär ist.

- b) Es sei  $\Sigma = \{0, \dots, 9\}$  und  $L \subseteq \Sigma^*$  die Menge aller Dezimaldarstellungen von Zahlen, die durch 2009 teilbar sind. Begründen Sie unter Verwendung des aus der Grundschule bekannten schriftlichen Divisionsalgorithmus, dass  $L$  regulär ist.

**Aufgabe 3:**

Alle Sprachen dieser Aufgabe sind über dem Alphabet  $\Sigma = \{a, b, c\}$  definiert.

- a) Beweisen oder widerlegen Sie folgende Aussage: Sind  $L_1$  und  $L_2$  kontextfrei, so auch ihre Differenz  $L_1 \setminus L_2$ .
- b) Zeigen Sie, dass die Sprache aller Wörter  $w$ , die die folgenden zwei Bedingungen erfüllen, kontextfrei ist:
  - Nach einem  $b$  kommt kein  $c$  mehr, also  $w = ubv \Rightarrow v \in \{a, b\}^*$ ;
  - Es sind nie mehr  $c$ 's als  $a$ 's, also  $w = uv \Rightarrow |u|_a \geq |u|_c$

Hier bedeutet  $|w|_x$  die Zahl der Vorkommen des Symbols  $x$  im Wort  $w$ .

**Aufgabe 4:**

Das Problem der Schulstundenplanung ist wie folgt formuliert:

Gegeben: Eine Menge  $L$  von Lehrern, eine Menge  $K$  von Klassen, eine Menge  $Z$  von Zeiten und eine Menge  $U$  von „Unterrichten“. Außerdem Funktionen  $k : U \rightarrow K$  und  $l : U \rightarrow L$ , die jedem Unterricht  $u \in U$  eine Klasse  $k(u)$  und einen Lehrer  $l(u)$  zuordnen.

Gesucht: Eine Funktion  $z : U \rightarrow Z$ , sodass für alle  $u, v \in U$  gilt:

$$u \neq v \wedge z(u) = z(v) \Rightarrow l(u) \neq l(v) \wedge k(u) \neq k(v)$$

d.h. gleichzeitig stattfindende Unterrichte gehören weder zum selben Lehrer, noch zur selben Klasse.

Zeigen Sie, dass das Problem der Schulstundenplanung NP-vollständig ist. Für die NP-Schwere können Sie z.B. auf Graphenfärbung reduzieren.

**Aufgabe 5:**

Es bezeichne  $F(h)$  die minimale Zahl von Knoten, die ein AVL-Baum der Höhe  $h$  enthalten kann. Die Höhe eines Baumes ist die Länge (Zahl der Kanten) des längsten Pfades von der Wurzel zu einem Blatt. Eine isolierte Wurzel hat Höhe 0. Eine Wurzel mit ein oder zwei Blättern als Kinder hat Höhe 1, etc.

- Geben Sie ein Beispiel eines AVL-Baums der Höhe 3 mit nur 7 Knoten und begründen Sie, dass kein AVL Baum der Höhe 3 mit weniger als 7 Knoten existiert, dass also  $F(3) = 7$  gilt.
- Begründen Sie ebenso, dass  $F(0) = 1, F(1) = 2, F(2) = 4, F(4) = 12$ .
- Leiten Sie aus diesen Beispielen eine Vorschrift ab, mit deren Hilfe man  $F(h+2)$  aus  $F(h)$  und  $F(h+1)$  berechnen kann.
- Begründen Sie die Allgemeingültigkeit dieser Vorschrift so genau wie möglich.

**Aufgabe 6:**

Die Wäscheleinaufgabe besteht darin,  $n$  Wäschestücke der Breiten  $b_1, b_2, \dots, b_n$  auf Wäscheleinen der Breite  $b$  aufzuhängen. Idealerweise sollte die Zahl der benutzten Leinen möglichst klein werden. Formal ist eine Aufhängung der Wäsche auf  $l$  Leinen also eine Einteilung der Menge  $\{1, \dots, n\}$  in  $l$  Klassen  $L_1, \dots, L_l$ , sodass für alle  $j = 1 \dots l$  gilt  $\sum_{i \in L_j} b_i \leq b$ . Eine Lösung der Wäscheleinaufgabe ist dann eine Zahl  $l$  und eine Aufhängung der Wäsche auf  $l$  Leinen. Eine Lösung ist umso besser, je kleiner  $l$  ist.

- Beschreiben Sie einen sinnvollen Greedy-Algorithmus für das Wäscheleinenproblem. (Also nicht einfach für jedes Wäschestück eine neue Leine)
- Geben Sie ein Beispiel einer Wäscheladung (Instanz des Wäscheleinenproblems), für die Ihr Algorithmus mehr als die minimal mögliche Zahl von Leinen verbraucht.
- Nennen Sie ein Beispiel einer Problemstellung, die mit einem Greedy-Algorithmus optimal gelöst werden kann.