

---

<b>Prüfungsteilnehmer</b>	<b>Prüfungstermin</b>	<b>Einzelprüfungsnummer</b>
---------------------------	-----------------------	-----------------------------

---

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

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

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

**Frühjahr  
2016**

**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:**      **6**

---

**Bitte wenden!**

**Thema Nr. 1****Aufgabe 1:** reguläre Mengen/Ausdrücke

Sei  $L$  die Sprache (Menge) aller Binärzahlen

- ohne führende Nullen,
- die durch 2 teilbar sind und
- die eine ungerade Anzahl an Nullen haben.

- a) Beschreiben Sie  $L$  durch einen regulären Ausdruck.
- b) Konstruieren Sie einen deterministischen endlichen Automaten (DFA)  $A$  für  $L$  (durch Angabe von  $A$  oder eines Diagramms für  $A$ ).

**Aufgabe 2:** kontextfrei

Zeigen Sie, dass die Sprache  $L = \{a^p b^q c^q \mid p, q > 0\}$

- a) kontextfrei ist.
- b) nicht regulär ist.

**Aufgabe 3:** regulär

Die Aussage: „jede Teilmenge einer regulären Menge ist regulär“ ist falsch.  
Begründen Sie dies.

**Aufgabe 4:** Turingmaschinen

Geben Sie eine Turingmaschine  $M$  an, die die Sprache  $\{w d w \mid w \in \{a,b,c\}^*\}$  akzeptiert.  
Beschreiben Sie in Worten, wie  $M$  arbeitet.

Welche Laufzeit hat  $M$  auf Inputs der Länge  $2n+1$  (oder  $n = |w|$ )?

**Aufgabe 5:** NP

Beschreiben Sie, was es heißt, dass ein Problem (Sprache) NP-vollständig ist.  
Geben Sie ein NP-vollständiges Problem Ihrer Wahl an und erläutern Sie, dass (bzw.) warum es in NP liegt.

**Fortsetzung nächste Seite!**

**Aufgabe 6: Datentypen**

Betrachten Sie das Verhalten von Kellern (stack), Warte-Schlangen (queue) und Prioritätswarteschlangen (heap).

Für die Prioritäten sind die Werte maßgeblich mit  $x$  vor  $y$  genau dann, wenn  $x > y$ .

$\text{in}(x)$  bedeutet, dass die Zahl  $x$  eingefügt wird.  $\text{out}()$  gibt eine Zahl aus.  
Am Anfang (und am Ende) sind alle Datentypen leer.

In welcher Reihenfolge werden die Zahlen bei der nachfolgenden Sequenz ausgegeben:

- a) bei einem Keller
- b) bei einer Warteschlange
- c) bei einer Prioritätswarteschlange (mit  $x > y$ )

$\text{in}(7), \text{in}(3), \text{in}(8), \text{in}(5), \text{in}(6), \text{out}(), \text{out}(), \text{out}(), \text{in}(2), \text{in}(1), \text{out}(), \text{out}(), \text{out}(), \text{in}(4), \text{out}(), \text{out}()$

**Aufgabe 7: O-Notation**

Gegeben sind die Funktionen (über den natürlichen Zahlen):

$$g(n) = 100 n \log n + 5 n + 10 \quad \text{und} \quad f(n) = n^3.$$

Zeigen Sie, dass  $g \in O(f)$  gilt.

(Es reicht nicht zu sagen, dass eine Funktion stärker steigt.)

**Aufgabe 8: Sortieren**

- a) Sortieren Sie das Array mit den Integer Zahlen

25, 1, 12, 27, 30, 9, 33, 34, 18, 16

- i) mit BubbleSort

- ii) mit Quicksort,

wenn als Pivotelement das jeweils erste Element gewählt wird.

Beschreiben Sie die Abläufe der Sortiervverfahren

- i) bei BubbleSort durch eine Angabe der Zwischenergebnisse nach jedem Durchlauf

- ii) bei Quicksort durch die Angabe der Zwischenergebnisse nach den rekursiven Aufrufen.

- b) Welche Laufzeit (asymptotisch, in O-Notation) hat BubbleSort bei beliebig großen Arrays mit  $n$  Elementen.  
Begründen Sie Ihre Antwort.

**Fortsetzung nächste Seite!**

**Aufgabe 9: AVL-Bäume**

Fügen Sie nacheinander die Zahlen 10, 18, 7, 14, 12, 19, 20 in einen anfangs leeren AVL-Baum ein.

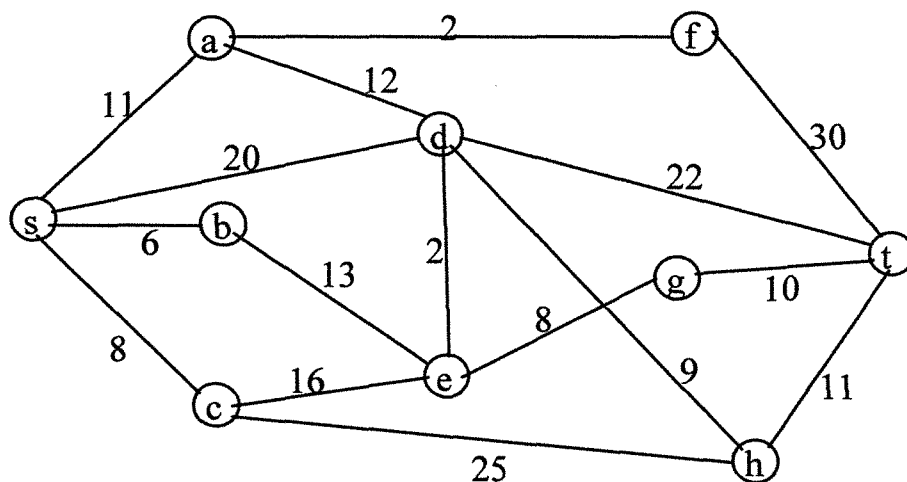
Repräsentieren (zeichnen) Sie alle AVL-Bäume jeweils vor einer notwendigen Rotation und beschreiben Sie die Rotationen (z.B. LL-Rotation für die Nummern ....).

**Aufgabe 10: minimaler Spannbaum**

Bestimmen Sie einen minimalen Spannbaum für den nachfolgend gezeichneten Graphen.

Zeichnen Sie die Spannbaumkanten ein.

Wie groß (Wert) ist der minimale Spannbaum?



**Thema Nr. 2****Aufgabe 1:**

Betrachten Sie die Sprache  $L$  aller Wörter über dem einelementigen Alphabet  $\Sigma = \{a\}$ , deren Länge eine Zweierpotenz ist, also  $L = \{a, aa, aaaa, \dots\}$ .

- a) Begründen Sie ausführlich, warum diese Sprache nicht kontextfrei ist.
- b) Betrachten Sie nun die folgende Grammatik  $G$ :

$$S \rightarrow a \mid SS$$

Begründen Sie  $L \subseteq L(G)$ , aber  $L \neq L(G)$ . Ordnen Sie  $L(G)$  bestmöglich in die Chomsky-Hierarchie ein und begründen Sie Ihre Antwort geeignet.

- c) Bringen Sie folgende Grammatik in Chomsky-Normalform und führen Sie sodann auf der Eingabe  $w = aabbb$  den Cocke-Younger-Kasami (CYK) Algorithmus durch.

$$S \rightarrow XY$$

$$X \rightarrow aXb \mid aXbb \mid ab$$

$$Y \rightarrow bY \mid b$$

**Aufgabe 2:**

Ein Raumausstattungsunternehmen steht immer wieder vor dem Problem, feststellen zu müssen, ob ein gegebener rechteckiger Fußboden mit rechteckigen Teppichresten ohne Verschnitt ausgelegt werden kann. Alle Längen sind hier ganzzahlige Meterbeträge. Haben sie beispielsweise zwei Reste der Größen  $3 \times 5$  und einen Rest der Größe  $2 \times 5$ , so kann ein Fußboden der Größe  $8 \times 5$  ausgelegt werden.

Das Unternehmen beauftragt eine Softwarefirma mit der Entwicklung eines Programms, welches diese Frage für beliebige Größen von Fußboden und Teppichresten entscheiden soll. Bei der Abnahme weist die Softwarefirma darauf hin, dass das Programm im wesentlichen alle Möglichkeiten durchprobiert und daher für große Eingaben schnell ineffizient wird. Auf die Frage, ob man das nicht besser machen könne, lautet die Antwort, dass das vorgelegte Problem NP-vollständig sei und daher nach derzeitigem Kenntnisstand der theoretischen Informatik nicht mehr zu erwarten sei.

- a) Fixieren Sie ein geeignetes Format für Instanzen des Problems und geben Sie konkret an, wie die obige Beispielinstantz in diesem Format aussieht.
- b) Begründen Sie, dass das Problem in NP liegt.
- c) Begründen Sie, dass das Problem NP-schwer (=NP-hard) ist durch Reduktion vom NP-vollständigen Problem SUBSET-SUM

**Fortsetzung nächste Seite!**

**Aufgabe 3:**

Sie sollen mithilfe von Falltests eine neue Serie von Smartphones auf Bruchsicherheit testen.

Dazu wird eine Leiter mit  $n$  Sprossen verwendet; die höchste Sprosse, von der ein Smartphone heruntergeworfen werden kann ohne zu zerbrechen, heiße "höchste sichere Sprosse". Das Ziel ist, die höchste sichere Sprosse zu ermitteln. Man kann davon ausgehen, dass die höchste sichere Sprosse nicht von der Art des Wurfs abhängt und dass alle verwendeten Smartphones sich gleich verhalten. Eine Möglichkeit, die höchste sichere Sprosse zu ermitteln, besteht darin, ein Gerät erst von Sprosse 1, dann von Sprosse 2, etc. abzuwerfen, bis es schließlich beim Wurf von Sprosse  $k$  beschädigt wird (oder Sie oben angelangt sind). Sprosse  $k-1$  (bzw.  $n$ ) ist dann die höchste sichere Sprosse. Bei diesem Verfahren wird maximal ein Smartphone zerstört, aber der Zeitaufwand ist ungünstig.

- a) Bestimmen Sie die Zahl der Würfe bei diesem Verfahren im schlechtesten Fall.
- b) Geben Sie nun ein Verfahren zur Ermittlung der höchsten sicheren Sprosse an, welches nur  $O(\log n)$  Würfe benötigt, dafür aber möglicherweise mehr Smartphones verbraucht.
- c) Es gibt eine Strategie zur Ermittlung der höchsten sicheren Sprosse mit  $O(\sqrt{n})$  Würfen, bei dessen Anwendung höchstens zwei Smartphones kaputtgehen. Finden Sie diese Strategie und begründen Sie Ihre Funktionsweise und Wurfzahl.

Tipp: der erste Testwurf erfolgt von Sprosse  $\lceil \sqrt{n} \rceil$ .