Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: ______	**Frühjahr 2010**	**66115**
Kennwort: ______		
Arbeitsplatz-Nr.: ______		

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

— Prüfungsaufgaben —

Fach: **Informatik**

Einzelprüfung: **Algorithmen und Datenstrukturen**

Anzahl der gestellten Themen (Aufgaben): 2

Anzahl der Druckseiten dieser Vorlage: 9

Bitte wenden!

Thema Nr. 1
(Aufgabengruppe)

Es sind alle Aufgaben dieser Aufgabengruppe zu bearbeiten!

Aufgabe 1:

a) Konstruieren Sie einen deterministischen endlichen Automaten für die durch den regulären Ausdruck $a(ba)^* + b + bb(ab)^*$ definierte Sprache über dem Alphabet $\Sigma = \{a, b\}$.

Minimieren Sie den Automaten mit dem Minimierungsalgorithmus oder weisen Sie von Ihrem Automaten nach, dass er bereits minimal ist.

b) Die Sprache $L = \{a^n b^n \mid n \geq 1\}$ über dem Alphabet $\Sigma = \{a, b\}$ ist bekanntlich kontextfrei. Ordnen Sie ihr Komplement $\Sigma^* \setminus L$ bestmöglich in die Chomskyhierarchie ein (mit detaillierter Begründung). Sie können sich dazu überlegen, welche Möglichkeiten es für ein Wort w gibt, die Mitgliedschaft in L zu verfehlen.

c) Wie man weiß, sind die kontextfreien Sprachen unter Schnitt mit regulären Sprachen abgeschlossen. Es ist auch wohlbekannt, dass die Sprache $\{a^n b^n c^n \mid n \geq 0\}$ nicht kontextfrei ist. Benutzen Sie diese Tatsachen (ohne Beweis!), um nachzuweisen, dass die Sprache $L = \{w \in \{a, b, c\}^* \mid |w|_a = |w|_b = |w|_c\}$ *nicht* kontextfrei ist. Erinnerung: $|w|_x$ bezeichnet für $x \in \{a, b, c\}$ die Anzahl der Symbole x in w.

Aufgabe 2:

Die Ackermannfunktion genügt bekanntlich den Gleichungen $a(0, y) = y + 1$ und $a(x + 1, 0) = a(x, 1)$ und $a(x + 1, y + 1) = a(x, a(x + 1, y))$ und wächst in beiden Argumenten streng monoton.

Eine Funktion $f : \mathbb{N} \to \mathbb{N}$ heiße *Ackermann-beschränkt*, wenn ein $k \in \mathbb{N}$ existiert, sodass $f(y) \leq a(k, y)$ für alle $y \in \mathbb{N}$. Zeigen Sie:

a) $a(1, y) = y + 2$ und $a(2, y) = 2y + 3$.

b) Sind f, g beide Ackermann-beschränkt, so auch $f + g$. Sie dürfen ohne Beweis verwenden: Für $x > 2$ gilt: $2a(x, y) \leq a(x + 1, y)$. Hinweis: $u + v \leq 2 \cdot \max(u, v)$.

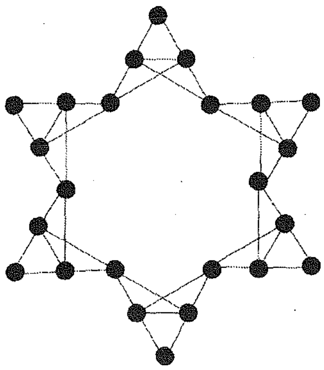
c) Ist f Ackermann-beschränkt, so auch die Funktion g, die durch $g(0) = 0$, $g(y + 1) = f(g(y))$ definiert ist (also kurz $g(y) = f^{(y)}(1)$).

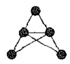
Fortsetzung nächste Seite!

Aufgabe 3:

Beim Graphenfärbungsproblem 3COL geht es darum, von einem gegebenen ungerichteten Graphen $G = (V, E)$ zu entscheiden, ob er mit drei Farben gefärbt werden kann, also ob eine Funktion $c : V \to \{0, 1, 2\}$ existiert, sodass aus $\{u, v\} \in E$ folgt $c(u) \neq c(v)$. Bekanntlich ist dieses Problem NP-vollständig.

a) Färben Sie den hier skizzierten Graphen mit drei Farben und begründen Sie dann, dass jede Färbung mit drei Farben den Zacken (die Knoten mit nur zwei Nachbarn) dieselbe Farbe zuweist.



b) Der obige Stern ist aus sechs Kopien eines "Eiffelturms"  zusammengesetzt. Begründen Sie, dass auch analog gebildete Sterne mit mehr als sechs Zacken auch mit drei Farben gefärbt werden können, aber wiederum nur so, dass alle Zacken gleichfarbig sind.

c) Das Problem 3COL4 ist die Einschränkung des Problems 3COL auf Graphen, bei denen jeder Knoten höchstens vier Nachbarn hat.

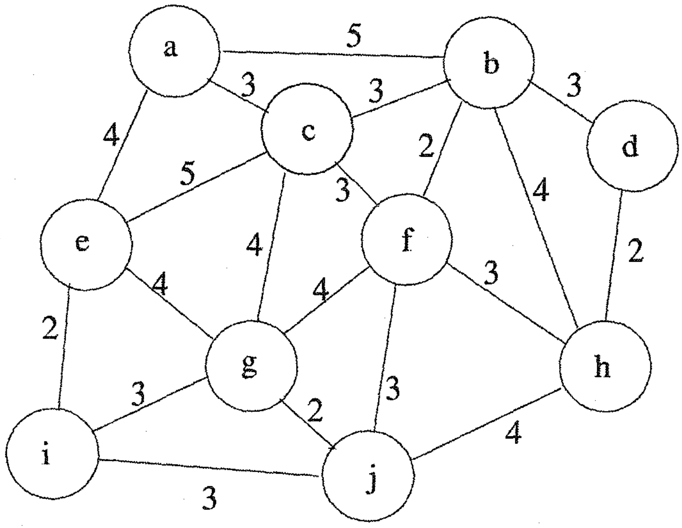
Zeigen Sie durch Angabe einer geeigneten Reduktion von 3COL auf 3COL4, dass bereits 3COL4 NP-schwierig (NP-hart) ist (mit Begründung). Idee: Knoten mit vielen Nachbarn durch Sterne wie in Teilaufgabe b) ersetzen.

Fortsetzung nächste Seite!

Aufgabe 4:

Professor Laksurk schlägt vor, minimale Spannbäume mit folgender Greedy-Strategie zu berechnen: Sortiere die Kanten nach absteigendem Gewicht, entferne solange Kanten in dieser Reihenfolge, wie der Graph (nach dem Entfernen) noch zusammenhängt.

a) Berechnen Sie zunächst mit einem der Standardalgorithmen einen minimalen Spannbaum des hier dargestellten Graphen. Dokumentieren Sie die Arbeitsschritte geeignet.



b) Berechnen Sie nunmehr einen minimalen Spannbaum mit Prof. Laksurks Verfahren. Die beiden Bäume sollten dasselbe Gewicht haben, denn die Strategie des Professors funktioniert tatsächlich!

c) Versuchen Sie nun die Korrektheit von Prof. Laksurks Strategie zu beweisen. Sie können hierfür die folgende Invariante einsetzen: "zu jedem Zeitpunkt enthält der noch vorhandene Graph einen minimalen Spannbaum des ursprünglichen Graphen". Dieser Teil ist vergleichsweise schwierig. Auch Ideen oder Teillösungen werden gewertet.

d) Mit welchem Algorithmus könnte man die Bedingung "der Graph hängt noch zusammen" überprüfen?

e) Geben Sie eine möglichst gute obere Schranke für die Laufzeit des Verfahrens unter Verwendung Ihrer Lösung von Teilaufgabe d) an. Verwenden Sie dabei die O-Notation. Vergleichen Sie die so ermittelte Laufzeit mit der eines Standardalgorithmus.

Thema Nr. 2
(Aufgabengruppe)

Es sind alle Aufgaben dieser Aufgabengruppe zu bearbeiten!

Aufgabe 1

Gegeben sei die Menge A:

> A sei die Menge der natürlichen Zahlen n, die bei der Division durch 7 den Rest 3 haben.
> $A = \{n \in \mathbb{N} \mid n \bmod 7 = 3\}$

a) Geben Sie eine Turing-Maschine TM ($TM=(Z, \Sigma, \Gamma, \delta, z_0)$) an, die die Menge A entscheidet. Die Überführungsfunktion können Sie als Tabelle oder Zustandsübergangsgraph darstellen.

Ermitteln Sie die Zeitkomplexität Ihrer TM.

b) Ist die Menge A semi-entscheidbar? Begründen Sie Ihre Entscheidung, indem Sie geeignete Änderungen an der Turingmaschine von a) vornehmen.

c) Beweisen Sie den folgenden Satz: Eine Menge A ist genau dann entscheidbar, wenn sowohl A als auch das Komplement von A semi-entscheidbar sind.

d) Beschreiben Sie das allgemeine Halteproblem. Begründen Sie informell, dass es nicht entscheidbar ist.

Aufgabe 2

Sind die folgenden Behauptungen zu regulären Ausdrücken wahr? Beweisen Sie Ihre Aussage.

a) $(R + S)^* = R^* + S^*$

b) $(R + RS)^*R = R(SR + R)^*$

Hinweis: $R^* = \bigcup_{i \geq 0} R^i$

Aufgabe 3

Die Funktion f sei eine berechenbare Funktion auf den natürlichen Zahlen mit $f(x) = 2x$.

a) Ist die Funktion f LOOP- oder WHILE-berechenbar?

b) Geben Sie ein LOOP- oder WHILE-Programm an, das die Funktion f berechnet.

Fortsetzung nächste Seite!

Aufgabe 4

Sei $\Sigma = \{0, 1, \$\}$, und sei $w \in \Sigma^*$. $\#_0(w)$ gibt an, wie oft die 0 in w vorkommt. Zum Beispiel ist $\#_0(0010) = 3$, $\#_0(01101) = 2$, $\#_0(1\$\$0\$) = 1$.

a) Sei

$$L_1 = \{\alpha\$\beta \mid \alpha, \beta \in \{0, 1\}^*, \#_0(\alpha) = \#_0(\beta)\} .$$

Beispiele: $11100\$1101011 \in L_1$, $\$ \in L_1$.

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

(a2) Beschreiben 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 Σ. Dann gibt . . .“

c) Zeigen Sie mittels Pumping-Lemma für kontextfreie Sprachen, dass die Sprache

$$L_2 = \{\alpha\$\beta\$\gamma \mid \alpha, \beta, \gamma \in \{0, 1\}^* , \#_0(\alpha) = \#_0(\beta) = \#_0(\gamma)\}$$

nicht kontextfrei ist.

Beispiele: $0100\$011010\$11000 \in L_2$, $00\$00\$00 \in L_2$

Aufgabe 5

Sind folgende Aussagen *richtig* oder *falsch*?

a) Die Suche nach einem Element in der Datenstruktur Keller/Stapel hat bei n Elementen den Aufwand $O(n)$.

b) Um n Elemente zu sortieren, braucht das Sortieren durch Zerlegen (Quicksort) zusätzlichen Speicherplatz in der Größenordnung von (log n).

c) Die Haldensortierung (Heapsort) hat einen niedrigeren Speicherplatzbedarf als das Sortieren durch Mischen (Mergesort).

d) Ein Blattbaum kann auch dann die AVL-Eigenschaft erfüllen, wenn er nicht balanciert ist.

e) Das Sortieren durch Mischen (Mergesort) hat für das Sortieren einer Reihung von n Zahlen im schlechtesten Fall einen Aufwand von $O(n^2)$.

f) Das Sortieren durch Zerlegen (Quicksort) hat für das Sortieren einer Reihung von n Zahlen im schlechtesten Fall einen Aufwand von $O(n^2)$.

Fortsetzung nächste Seite!

Aufgabe 6

Gegeben sei eine Reihung `a[i]` (`i = 0,1,...,n-1`) gefüllt mit paarweise verschiedenen ganzen Zahlen.

Ergänzen Sie folgendes Programmfragment so, dass die Elemente mittels Quicksort aufsteigend sortiert werden. Geben sie dazu den Code an, der in die Bereiche `Ergaenzung 1` und `Ergaenzung 2` einzusetzen ist.

```
// vertauscht zwei Positionen einer Reihung
static void swap (int a[], int i, int j){
    int t=a[i]; a[i]=a[j]; a[j]=t;
}
// sortiert die komplette Reihung
static void quickSort (int a[]){
    quickSort(a, 0, a.length - 1);
}
// sortiert den Bereich von start bis end (inklusive)
static void quickSort (int a[] , int start , int end) {
    // Ergaenzung 1
    ...

    int pivot = a[start];
    int i = start ;
    int k = end;
    while (i < k) {
        while (i<k && a[i] < pivot) i++;
        while (i<k && a[k] > pivot) k--;
        if (i < k) swap (a, i, k);
    }
    // Ergaenzung 2
    ...
```

Aufgabe 7

Gegeben ist folgender Ausschnitt der Implementierung einer Datenstruktur X:

```
public class E {
      E v1 = null;
      E v2 = null;
      int w;
}

public class X {
      E e;

      public boolean m1() {
            return e == null;
      }

      public int m2() {
            if (m1()) {
                  return 0;
            } else {
                  return e.w;
            }
      }

      public int m3() {
            int i = 0;
            E e1 = e;
            while (e1 != null) {
                  e1 = e1.v2;
                  i++;
            }
            return i;
      }

      public void m4(int j) {
            (if !m1()) {
                  E e2 = new E();
                  e2.v2 = e;
                  e2.w = j;
                  e = e2;
            } else {
                  e = new E();
                  e.w = j;
            }
      }
}
```

a) Erläutern Sie, um welche Datenstruktur es sich hier handelt und welche Funktionalität die Methoden m1(), m2(), m3() und m4(int j) besitzen.

b) Beschreiben Sie in Stichworten unter Verwendung von veranschaulichenden Skizzen, wie beim Löschen eines ersten Vorkommens eines Elements mit einem bestimmten Wert w1 aus dieser Datenstruktur prinzipiell vorgegangen werden muss.

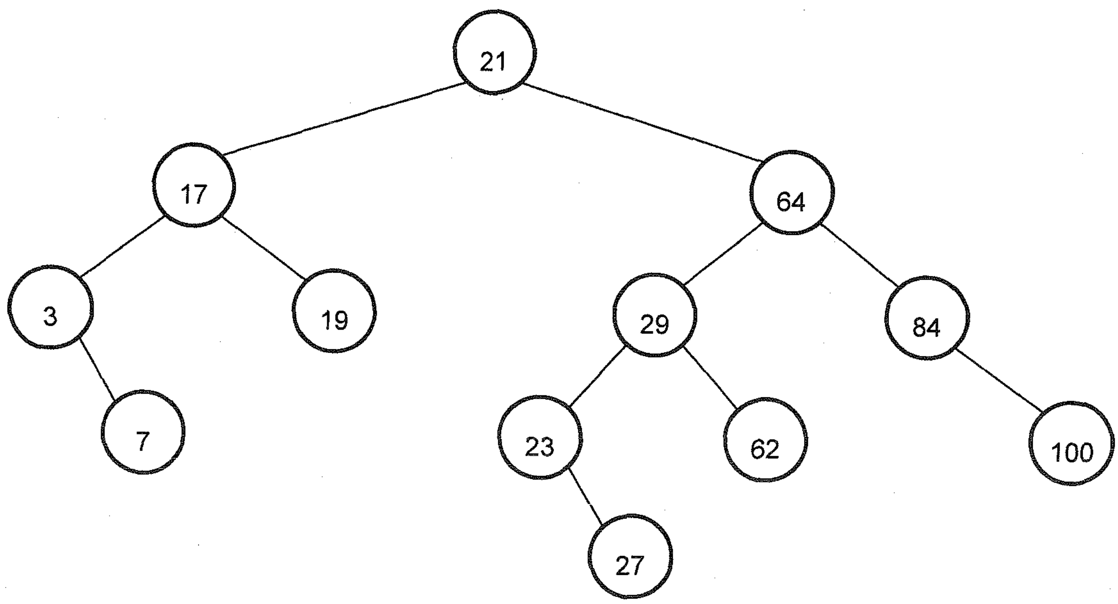
c) Implementieren Sie eine Methode, welche die in Teilaufgabe b) beschriebene Funktionalität besitzt. Verwenden Sie dabei eine gängige objektorientierte Programmiersprache oder einen entsprechenden Pseudocode und kommentieren Sie Ihre Lösung.

Fortsetzung nächste Seite!

Aufgabe 8

a) Erläutern Sie in Stichpunkten die Vor- und Nachteile der Datenstruktur AVL-Baum (jeweils mit Begründung).

b) Geben Sie für den folgenden AVL-Baum für alle Knoten jeweils die Balancierung an.



c) Erläutern Sie, ob es sich bei dem Baum aus Teilaufgabe b) immer noch um einen AVL-Baum handelt, wenn der Knoten mit dem Wert 62 entfernt wird. Skizzieren Sie gegebenenfalls erforderliche Maßnahmen, um den nach der Löschung des oben genannten Knotens entstandenen Baum wieder in einen AVL-Baum zu überführen.

d) Geben Sie die Folge der Knoten an, wenn der in Aufgabe b) gegebene Baum in preorder-Reihenfolge traversiert wird.