

**Aufgabe 0** Beschriftung

1 Punkt

Schreiben Sie auf **jedes** Blatt Ihren Namen und Ihre Matrikelnummer.

**Aufgabe 1** Rekurrenzen

7+5=12 Punkte

a) Lösen Sie die folgende Rekurrenz mit der Methode der generierenden Funktionen:

$$\begin{aligned} g_0 &= 1 \\ g_n &= 3 \cdot g_{n-1} + 2 \end{aligned}$$

b) Geben Sie für die folgenden Rekurrenzen eine geschlossene Form an.

$$1. \quad g_n = \begin{cases} 0 & : n \leq 0 \\ 3g_{n-1} + 5g_{n-2} + 7g_{n-3} & : n > 0 \end{cases}$$

2. Bei einer Rekurrenz sei der erste Wert  $g_0$  gleich eins und für alle weiteren Werte sei das Verhältnis zweier aufeinanderfolgender Werte konstant.

**Aufgabe 2** Aufwandsanalyse

6+2+5=13 Punkte

Betrachten Sie die folgende Funktion:

```
potenz x n k | n == 0           = 1
              | (n 'mod' k) == 0 = pot k (potenz x (n 'div' k) k)
              | otherwise       = x*(potenz x (n-1) k)
```

```
where pot 0 y = 1
```

```
pot i y = y * pot (i-1) y
```

a) Schätzen Sie die Anzahl der Multiplikationen von `potenz x n k` für eine beliebiges aber festes  $k > 1$  mit dem  $\Theta$ -Kalkül ab. Gehen Sie vom schlimmsten Fall aus und begründen Sie Ihre Antwort.

b) Wieviele Divisionen (`mod, div`) braucht `potenz x n n`? Schätzen Sie die Anzahl für den besten Fall mit dem  $\Omega$ -Kalkül ab.

c) Zeigen Sie mit vollständiger Induktion über  $n$ :  
Für  $n > 0$  und  $k > 1$  gilt: `potenz x n k = xn`

**Aufgabe 3** Algorithmen

6+3+3=12 Punkte

Gegeben sei ein Element  $y$  und eine Reihung  $A$  mit  $n$  Elementen mit der Eigenschaft:

Für alle  $i, j \in \{1, \dots, n\}$  gilt  $i < j \Rightarrow A[i] \leq A[j]$ .

a) Finden Sie einen möglichst effizienten Algorithmus, um  $y$  in der Reihung  $A$  zu finden. Schätzen Sie den Aufwand Ihres Algorithmus mit dem  $\Theta$ -Kalkül ab. Begründen Sie Ihre Schätzung.

- b) Nun soll  $y$  gelöscht werden und ein neues Element  $x$  eingefügt werden. Welchen Aufwand verursacht dies im schlimmsten Fall, wenn die obige Eigenschaft von  $A$  beibehalten werden soll? Begründen Sie Ihre Antwort.
- c) Gegeben sei eine Hashtabelle mit linearer Sondierung. Was ist eine Kollision zweiter Ordnung? Geben Sie ein Beispiel an.

**Aufgabe 4** Logik

2+2+5+3=12 Punkte

- a) Was ist ein Unifikator?
- b) Ist der allgemeinste Unifikator  $mgu$  immer eindeutig bestimmt? Beweisen Sie Ihre Antwort.
- c) Zeigen Sie, daß für eine PL-Formel  $F$  folgende Aussage gilt:  
Sind  $F$  und  $\neg F$  erfüllbarkeitsäquivalent, so ist  $F$  erfüllbar.  
Gilt auch die Umkehrung? Beweisen Sie Ihre Antwort.
- d) Gegeben sei ein zweistelliges Prädikatsymbol  $P$ . Geben Sie eine Formel  $F$  derart an, daß für jedes Modell  $\mathcal{A} = (U_{\mathcal{A}}, I_{\mathcal{A}})$  von  $F$  gilt:  $I_{\mathcal{A}}(P)$  ist eine transitive Relation.

**Aufgabe 5** Multiple Choice

10 Punkte

Markieren Sie in der folgenden Tabelle, ob die Aussage richtig oder falsch ist. Beachten Sie dabei: Für jede richtige Antwort gibt es einen Punkt und für jede falsche Antwort wird ein Punkt abgezogen. Die minimale Punktzahl der Aufgabe ist Null.

Nr.	Behauptung	richtig	falsch
1.	$xRy$ gdw. $x \in \Theta(y)$ ist eine Äquivalenzrelation.		
2.	Selection-Sort braucht weniger Vertauschungen als Merge-Sort.		
3.	Für zwei prädikatenlogische Formeln $F$ und $G$ gilt: Ist $F \rightarrow G$ eine Tautologie, so sind $F$ und $G$ erfüllbarkeitsäquivalent.		
4.	Besitzt ein Graph eine transitive Kante, so ist er kein Baum.		
5.	Die Höhe eines Suchbaums mit $n$ Schlüsseln liegt in $O(\log n)$ .		
6.	$O(n^2) + \Theta(n) = O(n^2)$		
7.	Die Suche des Wortes "gefunden" in beliebigen Texten der Länge $n$ ist mit dem Algorithmus von Knuth, Morris und Pratt mit $O(n)$ Vergleichen möglich.		
8.	Löschen des größten Elements in einem Heap verursacht konstanten Aufwand.		
9.	Wenn zwei PL-Formeln $F, G$ keine Modelle besitzen, so sind sie kongruent.		
10.	In der objektorientierter Programmierung gilt: Jede Klasse kennt die von ihr abgeleiteten Objekte.		