Probeklausur der Tutoren

Name:

Matrikelnr.:

Aufgabe	1	2	3	4	5	6	Summe
Maximal	6	9	10	12	9	14	60
Erreicht							

Aufgabe 1: Verständnis- und Wissensfragen (6 Punkte)

Kreuzen Sie an, ob die Aussage wahr ($\boxed{\text{W}}$) oder falsch ($\boxed{\text{F}}$) ist.
Hinweis: Jedes korrekte Kreuz zählt 0,5 Punkte, jedes falsche Kreuz bewirkt 0,5 Punkte Abzug! Die Teilaufgabe wird mindestens mit 0 Punkten bewertet.

W	F	Bei der Konstruktion eines Huffmanbaums vereinigt der Greedy-Algorithmus zuerst die großen Häufigkeiten, damit diese dann weit oben im Baum stehen.
W	F	Es gibt prädikatenlogische Formeln, die keine äquivalente Formel in konjunktiver Normalform besitzen.
W	F	Zu jeder prädikatenlogischen Formel gibt es eine äquivalente Formel in Skolemnormalform.
W	F	Zu jeder Häufigkeitsverteilung eines Alphabets, ist der Huffmanbaum und sind damit die Huffmancodeworte eindeutig bestimmt.
W	F	Für alle Graphen G gilt: G ist antisymmetrisch $\Leftrightarrow \neg(G$ ist symmetrisch)
W	F	Für alle Graphen G gilt: G ist zyklisch $\Leftrightarrow \neg(G$ ist azyklisch)
W	F	Beim O-Kalkül gilt: $g(n) \in O(f(n)) \Rightarrow f(n) \in O(g(n))$
W	F	Monte-Carlo-Algorithmen terminieren immer.
W	F	Der Algorithmus von Pollard-Rho berechnet die Primfaktorzerlegung einer natürlichen Zahl.
W	F	Es gilt: $O(0) = O(1)$
W	F	Rot-Schwarz-Bäume sind immer perfekt ausbalanciert.
W	F	$\Theta(f(n)) = \Theta(g(n)) \Longleftrightarrow f(n) \in o(g(n)) \wedge g(n) \in \omega(f(n))$.

Aufgabe 2: Relationen & Graphen (3 + 1 + 5 Punkte)

(a) Wieviele Kanten kann ein *irreflexiver* gerichteter Graph mit n Knoten höchstens haben. Beweisen Sie ihre Vermutung mittels Induktion.

Hinweis: Irreflexiv ist nicht dasselbe wie *nicht reflexiv.*

(b) Wieviele Kanten kann ein gerichteter Graph mit n Knoten höchstens haben, wenn er weder *reflexiv* noch *transitiv* ist.

(c) *Defintion:* Ein binärer Baum heißt Bruder-Baum, wenn

(i) jeder innere Knoten 1 oder 2 Nachfolger hat,
(ii) jeder unäre Knoten einen binären Bruder hat,
(iii) alle Blätter dieselbe Tiefe haben.

Wie viele Blatt-Knoten hat ein Bruder-Baum der Höhe 4 (Wurzel: Höhe 0, Blätter werden nicht gezählt), falls er eine minimale Anzahl von Blatt-Knoten hat?

Aufgabe 3: O-Kalkül (6 + 4 Punkte)

(a) Prüfen Sie, in welche der angeführten Komplexitätsklassen die *worst-case* Laufzeitkomplexität der folgendenen Algorithmen fällt.

Hinweis: Tragen Sie in jedes Kästchen entweder ein ✓ falls der Algorithmus in der Klasse ist, oder ein ✘ falls dieser nicht in der Klasse enthalten ist. Jeder korrekte Haken oder Kreuz zählt 0,25 Punkte, jeder falsche Haken oder Kreuz bewirkt 0,5 Punkte Abzug! Jeder der Algorithmen wird mindestens mit 0 Punkten und maximal mit 1 Punkt bewertet.

	$\omega(1)$	$\Theta(\log n)$	$O(n)$	$o(n \log n)$	$O(n^2)$	$\Omega(n^2)$	$O(n^3)$
Insertionsort	☐	☐	☐	☐	☐	☐	☐
Mergesort	☐	☐	☐	☐	☐	☐	☐
Quicksort	☐	☐	☐	☐	☐	☐	☐
Floyd-Warshall	☐	☐	☐	☐	☐	☐	☐
Rot-Schwarz-Baum-Insert	☐	☐	☐	☐	☐	☐	☐
Lookup in einem Hashtable	☐	☐	☐	☐	☐	☐	☐

(b) Beweisen Sie für zwei Funktionen $f, g : \mathbb{N} \to \mathbb{N}$ anhand der Definition des O-Kalküls: $\max\{f, g\} \in \Theta(f + g)$.

Aufgabe 4: Haskell (5 + 7 Punkte)

(a) Gegeben sei eine duplikatfreie Liste `l :: [a]`. Wir bezeichnen die Liste aller möglichen Teillisten mit Auslassungen als Kombinationen `c :: [[a]]` von `l`. Zum Beispiel sind die Kombinationen von `[1, 2, 3]` gerade `[ [1,2,3], [1,2], [1,3], [2,3], [1], [2], [3], [] ]`. Die relative Position der Kombination ist dabei unerheblich.

Schreiben Sie eine Funktion `combs :: [a] -> [[a]]`, welche die Kombinationen ihres Arguments berechnet.

Hinweis: Sie können dabei beliebige Funktionen der Standardbibliothek verwenden.

(b) Der ADT Menge modelliert eine Menge im mathematischen Sinne. Vervollständigen Sie das Modul Menge indem Sie die Operationen implementieren.

Hinweis: Eine Menge im mathematischen Sinne darf insbesondere keine zwei gleichen Elemente enthalten!

```
module Menge where

type Menge a = [a]

leereMenge    :: (Eq a) => Menge a                         -- liefert eine leereMenge
istLeer       :: (Eq a) => Menge a -> Bool                 -- Menge leer?
hatElement    :: (Eq a) => Menge a -> a -> Bool            -- Element in der Menge enthalten?
einfuegen     :: (Eq a) => Menge a -> a -> Menge a         -- fügt Element in eine Menge ein
vereinigung   :: (Eq a) => Menge a -> Menge a -> Menge a  -- vereinigt zwei Mengen
schnitt       :: (Eq a) => Menge a -> Menge a -> Menge a  -- schneidet zwei Mengen
```

Aufgabe 5: Prädikatenlogik (5 + 4 Punkte)

(a) Gegeben sei die prädikatenlogische Formel $F = \neg\Big(\exists z\big(P(z) \wedge \forall y(Q(y) \rightarrow \forall x\ R(x, y, z))\big)\Big)$.
Bereinigen Sie zunächst die Operatoren. Stellen Sie dann die bereinigte Pränexform her und erstellen Sie die Skolemform von F.

(b) Über den natürlichen Zahlen (ohne Null) seien folgende Prädikate definiert:

$$P(x, y) := \{(x, y) \mid x \text{ teilt } y \text{ ohne Rest}\}$$
$$Q(x, y) := \{(x, y) \mid x = y\}$$
$$R(x, y) := \{(x, y) \mid x < y\}$$

Formulieren Sie die folgenden Aussagen in Prädikatenlogik.

- x ist eine Primzahl
- x ist eine gerade Zahl
- x ist der ggT von y und z
- x und y sind teilerfremd

Aufgabe 6: Rekurrenzrelationen (6 + 4 + 4 Punkte)

(a) Gegeben sei folgende Haskellfunktion (1,5+1+1,5+2 Punkte):

```
s [x] = [x]
s xs  = min:s(o min xs)
        where min = minimum xs
              o x (y:ys) | x==y      = ys
                         | otherwise = y:(o x ys)
```

(i) Was macht die obige Funktion?

(ii) Welche Eingabe stellt den Worst-Case bezogen auf die Anzahl der Vergleiche dar?

(iii) Leiten Sie eine Rekurrenz für die Anzahl der Vergleiche her.
Hinweis: Nehmen Sie an, `minimum` habe einen Aufwand von n bei einer Eingabelänge von $n + 1$.

(iv) Lösen Sie die Rekurrenz.

(b) Gegeben seien

- die ersten k Glieder einer Rekurrenz: $f_0,\ f_1, \ldots, f_{k-1}$,
- die Differenzen dieser Glieder: $d_i = f_i - f_{i-1}$
- und deren Differenzen $D_i = d_i - d_{i-1}$.

Weiterhin gelte: $D_{i+1} = d_i$ für alle $i \geq 0$. In welcher Komplexitätsklasse liegt f? Begründen Sie Ihre Meinung.

(c) Finden Sie aus den Angaben von Teil (b) eine rekursive Formel für die Rekurrenz.