Prof. Dr. B. Baumeister
Dipl.-Math. Peter Knorr

WS 2009/10

Mathematik für Informatiker I - Übungsblatt 4

Abgabe bis Montag, den 9. November, 12 Uhr
in die Briefkästen im Mathe-Foyer

Aufgabe 1: (4 Punkte)

Die *Gaußklammer*-Funktion [ ] weist jedem Wert $x \in \mathbb{R}$ den Wert $k \in \mathbb{Z}$ zu, der durch Entfernung aller Nachkommastellen von x entsteht.
(Beispiele: $[1.88297] = 1,\ [-17.3927] = -17,\ [4] = 4$).
Die Relation $\sim$ sei durch $x \sim y :\Leftrightarrow [x] = [y]$ definiert.

(i) Beweisen Sie, dass $\sim$ eine Äquivalenzrelation ist.
(ii) Welche Gestalt haben die Äquivalenzklassen von $\sim$?
(iii) Konstruieren Sie ein Repräsentantensystem von $\sim$.

Aufgabe 2: (4 Punkte)

M sei die Menge $\{a, b, c, d, e, f\}$. Ergänzen Sie

(i) die Menge $\{(e, e), (f, d), (c, a), (b, f)\}$ zu einer Äquivalenzrelation auf M mit mindestens zwei Äquivalenzklassen,
(ii) die Menge $\{(a, a), (f, e), (c, d), (a, b), (c, a), (e, b), (f, d)\}$ zu einer totalen Ordnung auf M.

Aufgabe 3: (4 Punkte)

Beweisen Sie mit vollständiger Induktion, dass für $n \in \mathbb{N}_0$ ist die n-te Fibonacci-Zahl $F(n)$ gegeben ist durch die Formel
$F(n) = \frac{1}{\sqrt{5}}\left(\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right)$.

(Bemerkung: Dies ist Lemma 2.1.9; diese Aufgabe soll den Beweis ersetzen, der im Skript ausgelassen wurde. Verwenden Sie nicht das Lemma selbst!)

Bitte wenden!

Aufgabe 4: (4 Punkte)

Sei $F(n)$ die n-te Fibonacci-Zahl. Zur konkreten Berechnung von $F(n)$ zu einem bekannten n gibt es zwei Algorithmen:

(Algorithmus A)

Man führt `fibonacci(n)` durch. Die Programmroutine `fibonacci(k)` bekommt als Eingabevariable k und liefert $F(k)$ zurück. Ihr Ablauf:

- Wenn $k = 0$, dann $F(k) := 0$
- Wenn $k = 1$, dann $F(k) := 1$
- Wenn $k \notin \{0, 1\}$, dann:
  - Führe `fibonacci(k-1)` durch; erhalte $F(k-1)$
  - Führe `fibonacci(k-2)` durch; erhalte $F(k-2)$
  - Berechne $F(k) = F(k-1) + F(k-2)$
- Liefere den Wert $F(k)$ zurück.

(Algorithmus B)

1. Starte mit $F(0) = 0$, $F(1) = 1$, $k = 0$
2. Setze $G(k) = (F(k+1), F(k))$
3. Speichere $G(k)$
4. Berechne $F(k+2)$ als Summe der beiden Komponenten von $G(k)$
5. Erhöhe k um 1
6. Abbruch für $k + 1 = n$, sonst zurück zu Schritt 2

(i) Berechne die Anzahl der benötigten Additionen für beide Algorithmen am Beispiel $n = 6$.

(ii) Finde für beide Algorithmen eine allgemeine Formel für die Anzahl der Additionen in Abhängigkeit von n.