Prof. Andreas Goerdt
M. Pippig, M. Riedel, U. Schönke

TU Chemnitz
Wintersemester 2003/2004

Algorithmen und Programmierung

13. Übung

1. Aufgabe:

   Gegeben sei ein Feld $feld$ der Länge n, wobei $feld[i] \in \{0,\ldots,n-1\}$ mit $0 \leq i < n$ ist. Schreiben Sie eine Methode, die diejenigen Elemente ausgibt, die genau zwei mal im Feld vorkommen. Ihre Methode soll eine Zeit von $O(n)$ haben.

2. Aufgabe:

   Gegeben sei folgendes Programmstück:

```
// Eingabe x>=0
z = x;
y = 0;

while (x > 0) {
  y = y + z;
  x = x - 1;
}
```

   a) Was berechnet dieses Programmstück?

   b) Weisen Sie die Korrektheit Ihrer Aussage nach. Gehen Sie dabei nach dem Prinzip

   - Aussage über die Endlichkeit der Anzahl der Schleifendurchläufe
   - Beweis einer geeigneten Invariante
   - Quintessenz

   vor.

3. Aufgabe:

   Gegeben sei folgendes Programmstück:

```
//Eingabe x>=0, y>=0

while (x > 0) {
  y = y + 2;
  x = x - 1;
}
```

   a) Was berechnet dieses Programmstück?

b) Weisen Sie die Korrektheit Ihrer Aussage nach. Gehen Sie dabei wie bei Aufgabe 2 vor.

4. Aufgabe:

   Gegeben sei folgendes Programmstück:

```
//Eingabe x>=0
y = 0;

while (x > 0) {
  y = y + x;
  x = x - 1;
}
```

   a) Was berechnet dieses Programmstück?

   b) Weisen Sie die Korrektheit Ihrer Aussage nach. Gehen Sie dabei wie bei Aufgabe 2 vor.

5. Aufgabe:

   Die folgenden 32-stelligen Bitfolgen seien Darstellungen von Zahlen vom Typ `float` nach dem Standard IEEE 754 (siehe Vorlesung und 6. Übung). Welchen Dezimalzahlen entsprechen sie jeweils?

   a) 10111110100000000000000000000000

   b) 01000001010111000000000000000000

   c) 11000010000010000000000000000000

6. Aufgabe:

   Gegeben seien folgende periodische Binärzahlen:

   a) $1,\overline{01}$

   b) $1,\overline{001}$

   Berechnen Sie deren Wert im Dezimalsystem unter Verwendung der Formel für den Grenzwert der geometrischen Reihe:

$$\lim_{n\to\infty}\sum_{i=0}^{n} q^i = \frac{1}{1-q} \qquad \text{für } |q|<1, q\neq 0$$

7. Aufgabe:

   In der 6. Übung (1. Aufgabe) sollte ein effizienter Algorithmus entwickelt werden, der für ein $n>0$ testet, ob $n=a^b$ mit $a\geq 2$ und $b\geq 2$ ist. Der entwickelte Algorithmus arbeitete unter Verwendung der binären Suche (siehe Lösungsunterlagen). Welchen Aufwand (O-Notation) hat dieser Algorithmus?