



# Informatik II: Algorithmen und Datenstrukturen

SS 2002

## 7. Übungsblatt

### Aufgabe 21 (Fibonacci-Zahlen (2), 4P)

Implementieren Sie die beiden Varianten zur Berechnung der Fibonacci-Zahlen aus Aufgabe 18 (rekursiv und iterativ).

### Aufgabe 22 (Korrektheit für Multiplikation durch Addition, 5T)

In einer Sprache, in der für Zahlen  $a, b \in \mathbb{N}_0$  keine Multiplikation existiert, lässt sich diese durch folgende Funktion realisieren:

```
int mult(int x, int y)
{
    int ergebnis;
    ergebnis = 0;
    while (x != 0)
    {
        if (x % 2 == 1)
            ergebnis = ergebnis + y;
        x = x / 2;
        y = y + y;
    }
    return ergebnis;
}
```

Zeigen Sie, dass der Funktionsaufruf `mult(a,b)` für  $a, b \in \mathbb{N}_0$  das Produkt  $a \cdot b$  liefert. Fügen Sie hierzu hinter jeder Anweisung eine *assertion* ein.

**Hinweis:** Die Schleifeninvariante lautet

$$\{\mathcal{I} : \text{ergebnis} + x \cdot y = a \cdot b\}.$$

### Aufgabe 23 (Lineare Suche, 2T+6P)

Modifizieren Sie das Programm zur Linearen Suche aus der Vorlesung auf dreierlei Weise.

- Ändern Sie das Programm so, dass nicht mehr nach Zahlen, sondern nach Zeichenketten gesucht wird.
- Kommt die gesuchte Zahl unter den Datensätzen mehrfach vor, so liefert das Programm aus der Vorlesung das letzte Vorkommen zurück. Ändern Sie es so ab, dass statt dessen das erste Vorkommen zurückgegeben wird.
- Ändern Sie das Programm so, dass statt eines Datensatzes zurückgegeben wird, wie oft der gesuchte Schlüssel vorkommt. Wie groß ist hierfür der Aufwand im Best, Worst und Average Case?

**Abgabe:** Mi., 05.06.2002, 14 Uhr