

# Übungsblatt 6

Abgabe: 13.6.06 - 16.6.06

---

## Aufgabe 1 (Korrektheitsbeweise)

Punkte: 2+3

Geben Sie zu den unten stehenden Programmen je eine sinnvolle Schleifeninvariante an. Zeigen Sie damit die Korrektheit der Programme

a) Berechnung der  $n$ -ten Fibonacci-Zahl

```
F = [1,0]
i = 1
while i < n:
    F[1], F[0] = F[0], F[1] + F[0]
    i += 1
```

b) Innere Schleife von Bubblesort

```
i = 0
while i < k:
    if a[i] > a[i+1]:
        a[i], a[i+1] = a[i+1], a[i]
    i += 1
```

## Aufgabe 2 (Sortierverfahren)

Punkte: 2.5+2.5

a) Sortieren Sie die Zahlenfolge 2,4,8,6,1,5,3,7,9,0 mit Bubblesort, Insertionsort, Selectionsort, Shellsort( $h=5,3,1$ ) und Quicksort. Wählen Sie als Pivot-Element das Element einer Partition mit dem grössten Index. Geben Sie dabei die Zahlenfolge nach jedem äusseren Schleifendurchlauf bzw. nach jedem Rekursionsschritt an.

b) Zeigen Sie anhand eines Beispiels, daß Selectionsort, Shellsort und Quicksort nicht stabil sind. Warum sind Bubblesort und Insertionsort stabil?

## Aufgabe 3 (Bubblesort)

Punkte: 1+3+2

Bubblesort ist stark asymmetrisch bezüglich der Durchlaufrichtung, d.h. Elemente, die nicht in der richtigen Reihenfolge sind, werden am Anfang der Datenfolge schneller behoben als am Ende der Datenfolge.

a) Geben Sie eine zu sortierende Datenfolge mit 5 Elementen an, bei der diese Asymmetrie besonders deutlich wird. Begründen Sie kurz, wieso Ihre Datenfolge die Asymmetrie verdeutlicht.

b) Um die Asymmetrie zu vermeiden ändert man Bubblesort so, daß die Durchlaufrichtung wechselt, d.h. man geht abwechselnd von Links nach Rechts und von Rechts nach Links durch. Diese Variante heißt *Shakersort*. Implementierung Sie eine effiziente Version von Shakersort in Python.

c) Implementieren Sie eine verbesserte Version von Shakersort in Python, in der man die Schleifengrenzen dynamisch möglichst präzise anpasst