

## Informatik I: Einführung in die Programmierung

Prof. Dr. Bernhard Nebel  
Dr. Christian Becker-Asano, Dr. Stefan Wölfl  
Wintersemester 2014/2015

Universität Freiburg  
Institut für Informatik

### Übungsblatt 12

**Abgabe: Freitag, 30. Januar 2015, 18:00 Uhr**

**WICHTIGE HINWEISE:** Zur Bearbeitung der Übungsaufgaben legen Sie bitte ein neues Unterverzeichnis `sheet12` im Wurzelverzeichnis Ihrer Arbeitskopie des SVN-Repositories an. Ihre Lösungen werden dann entsprechend dem ersten Übungsblatt in Dateien in diesem Unterverzeichnis erwartet. Beachten Sie bitte bei allen Aufgaben die *Hinweise zur Bearbeitung der Übungsaufgaben* unter der folgenden URL:

[http://gki.informatik.uni-freiburg.de/teaching/ws1415/info1/hinweise\\_uebungen.txt](http://gki.informatik.uni-freiburg.de/teaching/ws1415/info1/hinweise_uebungen.txt)

Überprüfen Sie, dass Sie alle Lösungen ins Repository hochgeladen haben. Bewertet wird bei allen Aufgaben die letzte Version, die zur Deadline des Übungsblattes auf dem SVN-Server eingereicht ist.

**Aufgabe 12.1** (Komplexitätsanalyse; Punkte: 2+3+3+4; Dateien: `komplexitaet.txt`, `search.py`)

Betrachten Sie die beiden Funktionen `search1` und `search2`, die zusammen mit den Hilfsfunktionen `noop` und `fib` in folgender Datei definiert sind:

<http://www.informatik.uni-freiburg.de/~ki/teaching/ws1415/info1/python/search.py>

```
def noop(elm):
    return elm

def fib(elm):
    return elm if elm <= 1 else fib(elm-1) + fib(elm-2)

def search1(elm, lst, fct):
    for elm1 in lst:
        for elm2 in lst:
            if elm == fct(elm1 + elm2):
                return True
    return False

def search2(elm, lst, fct):
    length = len(lst)
    for i in range(length):
        for j in range(i, length):
            if elm == fct(lst[i] + lst[j]):
                return True
    return False
```

- (a) Erklären Sie, inwiefern sich die beiden Suchfunktionen prinzipiell unterscheiden. Sind sie semantisch äquivalent? Begründen Sie.
- (b) Schätzen Sie mit Hilfe der Landauschen  $\mathcal{O}$ -Notation die asymptotische Laufzeit der beiden Implementierungen in Abhängigkeit von der *Länge der Eingabeliste* `lst` ab. Für Ihre Analyse sollen Sie davon ausgehen, dass die Aufrufe der Funktion `fct` konstante Kosten aufweisen.
- (c) Im Folgenden wollen wir den Einfluß des Laufzeitverhaltens von `fct` empirisch testen: Integrieren Sie dazu eines der Templates vom vorherigen Aufgabenblatt 11 und passen Sie es so an, dass Sie die obigen Funktionen `search1` und `search2` auf etwaige Laufzeitunterschiede testen. Binden Sie dazu beide Hilfsfunktionen `noop` und `fib` nacheinander ein und vergleichen Sie. Diskutieren Sie die Ergebnisse.
- (d) Geben Sie für die unten angegebenen Funktionen  $a(n)$ ,  $b(n)$  und  $c(n)$  paarweise an, in welcher asymptotischen Beziehung sie zueinander stehen. Benutzen Sie dazu die Landausche  $\mathcal{O}$ -Notation und begründen Sie kurz.

$$a(n) = 2^n + n^3, \quad b(n) = 0.4^{2n}, \quad c(n) = 42^{1000}$$

**Aufgabe 12.2** (Lambda Notation; Punkte: 2+4; Datei: `lambda.txt`, `lambda.py`)

Gegeben sei folgende Liste `data`:

```
data = [('Bonn', '50.74 7.1 65'),
        ('Freiburg', 'BW', '48 7.84 270'),
        ('Bielefeld', 'NRW', '52.03 8.53 120')]
```

- (a) Welche Konsolenausgabe liefert der Aufruf `sorted(data)` und warum?
- (b) Modifizieren Sie den Aufruf der Funktion `sorted()` durch Spezifikation des optionalen Arguments `key=...`, so dass die Liste anhand der Höhenangaben der Städte absteigend (also z.B. 'Freiburg' vor 'Bielefeld') sortiert wird. Nutzen Sie dazu eine geeignete `lambda`-Funktion.

Ihre modifizierte Funktion sollte natürlich für beliebige Datensätze der o.a. Form eine korrekte Sortierung erzeugen, wobei Sie lediglich davon ausgehen können, dass die Zeichenkette mit den GPS-Koordinaten immer als letztes Element eines (beliebig langen) Tupels in der angegebenen Form erscheint und in dieser Zeichenkette die Höhe (über NN) als die dritte Zahl (wie im Beispiel) angeführt wird.

**Aufgabe 12.3** (Erfahrungen; Datei: `erfahrungen.txt`; Punkte: 2)

Legen Sie im Unterverzeichnis `sheet12` eine Textdatei `erfahrungen.txt` an. Notieren Sie in dieser Datei kurz Ihre Erfahrungen beim Bearbeiten der Übungsaufgaben (Probleme, benötigter Zeitaufwand nach Teilaufgabe, Bezug zur Vorlesung, Interessantes, etc.).