

Informatik I: Einführung in die Programmierung

Prof. Dr. Bernhard Nebel

Dr. Christian Becker-Asano, Dr. Stefan Wöflf

Wintersemester 2013/2014

Universität Freiburg

Institut für Informatik

Übungsblatt 11

Abgabe: Freitag, 31. Januar 2014, 18:00 Uhr

Bei diesem Übungsblatt erwarten wir alle Lösungen zu den Aufgaben in der jeweils angegebenen Datei im Unterverzeichnis `sheet11`. Beachten Sie dabei die bekannten Formatierungshinweise für Python-Dateien und vergessen Sie nicht, alle von Ihnen verwendeten Dateien ins SVN-Repository hochzuladen!

Aufgabe 11.1 (Komplexitätsanalyse; Punkte: 2+3+3+2; Dateien: `ex11-1.txt`, `search.py`)

Betrachten Sie die folgenden beiden Funktionen:

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

def search2(elm, lst):
    length = len(lst)
    for i in range(length):
        for j in range(i + 1, length):
            if elm == lst[i] + lst[j]:
                return True
    return False
```

- Erklären Sie, inwiefern sich diese beiden Funktionen prinzipiell unterschiedlich verhalten.
- Schätzen Sie mit Hilfe der Landauschen \mathcal{O} -Notation die asymptotische Laufzeit der beiden Funktionen in Abhängigkeit der Länge der Eingabeliste `lst` ab. Sind die asymptotischen worst-case Laufzeiten von `search1` und `search2` unterschiedlich? Begründen Sie.
- Nutzen Sie eines der templates vom vorherigen Aufgabenblatt 10. Speichern Sie es als `search.py` ab und modifizieren Sie die Datei dann so, dass Sie die obigen Funktionen `search1` und `search2` auf etwaige Laufzeitunterschiede testen. Diskutieren Sie das (überraschende?) Ergebnis.

- (d) Geben Sie für die unten angegebenen Funktionen $a(n)$, $b(n)$ und $c(n)$ paarweise an, wie sie bezüglich ihrer jeweiligen worst-case Laufzeiten in Beziehung zueinander stehen. Benutzen Sie dazu die Landausche \mathcal{O} -Notation und begründen Sie kurz.

$$a(n) = 3^n + n^2, \quad b(n) = 12^{2n}, \quad c(n) = 2^{1000}$$

Aufgabe 11.2 (Lambda Notation; Punkte: 1+3+4; Datei: `ex11-2.txt`, `lambda.py`)

Gegeben sei folgende Liste `data`:

```
data = [(4, 'Al Zero'), (1, 'John Doe'), (3, 77, 'Xavier Ant')]
```

- (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 Nachnamen der Personen (also z.B. 'Xavier Ant' vor 'John Doe') sortiert wird. Nutzen Sie dazu eine geeignete `lambda`-Funktion. (Speichern Sie Ihre Lösung in der Datei `lambda.py` ab.)
- (c) Schreiben Sie einen Einzeiler, also eine einzelne Zeile Code, mit der aus dem "Lorem ipsum"-Text `loremipsum_long` (vgl. Übungsblatt 5, Datei `words_data.py`) all jene Wörter herausgefiltert und als Menge auf der Konsole ausgegeben werden, die mehr als 333 mal im Text vorkommen. Nutzen Sie dazu die Funktion `filter` und eine geeignete `lambda`-Funktion. (Speichern Sie auch diese Lösung in der Datei `lambda.py` ab.)

Aufgabe 11.3 (Erfahrungen; 2 Punkte)

Legen Sie im Unterverzeichnis `sheet11` eine Textdatei `erfahrung.txt` an. Notieren Sie in dieser Datei kurz Ihre Erfahrungen beim Bearbeiten der Übungsaufgaben (Probleme, benötigter Zeitaufwand nach Teilaufgabe, Bezug zur Vorlesung, Interessantes, etc.).