



Übungen zu Computational Thinking

<http://www.mpi-inf.mpg.de/departments/d1/teaching/ws11/ct/>

Blatt 3

Abgabeschluss: 21. 11. 11 16:00

Regeln: Bis zum Semesterende müssen mindestens 42% der maximal erreichbaren Punkte aller Übungszettel erworben werden.

Programmcode ist elektronisch per E-Mail abzugeben. Zusätzlich müssen die Ausgaben einer exemplarischen Programmausführung mitgeliefert werden.

Aufgabe 1 (10 Punkte) Sortieren Sie die Wörter im Anhang nach dem Mergesort Verfahren.

Aufgabe 2 (10 Punkte) Sortieren Sie die Wörter im Anhang nach dem Bucketsort Verfahren.

Aufgabe 3 (10 Punkte) Finden Sie das Wort „Wiedereröffnungen“ in der Wortliste aus dem Anhang mit Binärsuche. Wieviele Vergleiche haben Sie gebraucht?

Aufgabe 4 (7 Punkte) Implementieren Sie die beiden in der Vorlesung vorgestellten Sortierverfahren, Sortieren durch Auswahl und Sortieren durch Mischen.

Aufgabe 5 (4 Punkte) Ein weiteres Sortierverfahren heißt „Sortieren durch Einfügen“. Implementieren Sie es an Hand der folgenden Schritte:

- Schreiben Sie eine Funktion, die ein Element so in eine sortierte Liste einfügt, dass die neue Liste wieder sortiert ist.
- Verwenden Sie die Funktion aus a) um die Elemente einer unsortierten Liste nacheinander in eine neue Liste einzufügen.
- Wieviele Vergleiche macht dieses Verfahren im schlimmsten Fall? Wieviele im günstigsten?

Aufgabe 6 (7 Punkte) Implementieren Sie das folgende rekursive Sortierverfahren:

```
sloSort(i, j, L)
  wenn L[i] > L[j] ist, vertausche die beiden Elemente
  wenn j-i > 1 ist,
    sei t = (j - i + 1) / 3
    sloSort(i, j-t, L)
```

```
sloSort(i+t, j, L)
sloSort(i, j-t, L)
```

Aufgerufen wird die Funktion mit $i=0$ und $j=\text{len}(L)$. Beschreiben Sie verbal, wie die Funktion sortiert. 5 *Bonuspunkte* erhalten Sie, wenn Sie bestimmen können, wieviele Vergleiche gebraucht werden.

Aufgabe 7 (12 Punkte) Ihre Tutoren verwalten die auf den Übungszetteln erreichten Punkte in einer Textdatei. Jede Zeile hat das Format

```
Vorname Nachname; Matrikelnummer; Punkte; Punkte; Punkte; ...
```

Um die Semikola können sich beliebig viele (Null eingeschlossen) Leerzeichen befinden. Das Benotungsschema kann sich noch ändern und ist daher in einer zweiten Textdatei angelegt. Die sieht beispielsweise so aus

```
95 1.0
90 1.3
84 1.7
78 2.0
72 2.3
66 2.7
60 3.0
54 3.3
48 3.7
42 4.0
```

Jede Zeile enthält `Mindestprozent Note`. Gehen Sie davon aus, dass jeder Übungszettel 60 Punkte wert ist. Schreiben Sie ein Programm, das nach den Pfaden zu den beiden Textdateien fragt und die Noten einmal nach Namen und einmal nach Matrikelnummern sortiert im Format `Name Matrikel Note` ausgibt.

Hilfreiche Funktionen sind unter anderem `split()` (mehr Informationen mit `help(str)`) und `sorted()` (siehe `help(sorted)`).

Aufgabe 8 (0 Punkte) Probieren Sie Ihre Sortierfunktionen aus und vergleichen Sie die Ausführungszeit auf verschiedenen Listen untereinander sowie mit der eingebauten Funktion `sorted`.

Wortliste:

'Megatonnen', 'therapeutischen', 'Wiedereintritt', 'genesener', 'reinitialisiere', 'Wiedereröffnungen', 'Ausmasse', 'dekontaminiertem', 'abnickten', 'anonymisiert', 'Profiverein', 'Distrikt', 'Heldengedichtes', 'geschriebener', 'aufzurufendes', 'dreimaligen', 'gefuehlsduseligen', 'ausgebliebenem', 'entlaste', 'abgekriegten'