



Übungen zu Computational Thinking

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

Blatt 2

Abgabeschluss: 14. 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 (5 Punkte) Kürzeste Wege. Schreiben sie auf, welche Transitknoten sie für Reisen von ihrer Heimatadresse aus benutzen.

Zur Erinnerung: jede Reise zu einem mehr als 100 Kilometer entfernten Ziel geht durch einen der Transitknoten.

Aufgabe 2 (15 Punkte) Stabile Paarungen. Vier Frauen und Männer haben die folgenden Präferenzen.

- **Fitzwilliam:** Elizabeth, Jane, Mary, Charlotte; **Charles:** Jane, Charlotte, Elizabeth, Mary; **George:** Jane, Elizabeth, Mary, Charlotte; **Bill:** Charlotte, Mary, Elizabeth, Jane;
- **Elizabeth:** Charles, George, Bill, Fitzwilliam; **Jane:** Charles, Bill, Fitzwilliam, George; **Charlotte:** Charles, Fitzwilliam, Bill, George; **Mary:** Charles, Bill, George, Fitzwilliam;

- a) Ist die Paarung (Fitzwilliam, Elizabeth), (Charles, Jane), (George, Charlotte), (Bill, Mary) stabil?
- b) Berechnen sie zwei stabile Paarungen nach dem Algorithmus aus der Vorlesung.
- i) Männer werben und Frauen wählen (so war es in der Vorlesung)
 - ii) Frauen werben, Männer wählen.
 - iii) Diskutieren sie die Unterschiede in den berechneten Paarungen

Aufgabe 3 (5 Punkte) Paarungen von größtem Nutzen. Drei Aufgaben sind von drei Personen zu erledigen. Die Personen sind unterschiedlich geeignet für die verschiedenen Aufgaben. Nehmen sie an, dass Person A 1 Stunden für die erste Aufgabe, 2 Stunden für die zweite Aufgabe und 3 Stunden für die dritte Aufgabe braucht. Bei Person B sind es 1, 1, und 2 Stunden, bei Person C sind es 0.5, 2, und 0.5 Stunden.

Ordnen Sie den Personen Aufgaben zu, so dass:

- a) Die Gesamtbearbeitungszeit minimal ist. Einer Person dürfen mehrere Aufgaben zugeordnet werden.
- b) Die Gesamtbearbeitungszeit minimal ist. Einer Person darf nur eine Aufgabe zugeordnet werden.
- c) Die letzte Aufgabe möglichst früh abgeschlossen wird. Einer Person dürfen mehrere Aufgaben zugeordnet werden.

Aufgabe 4 (0 Punkte) Lesen Sie die ersten vier Kapitel des Python Tutorials <http://docs.python.org/tutorial/index.html>.

Aufgabe 5 (0 Punkte) Probieren Sie die Funktion `printAll` aus den Tutoriumsnotizen für 998 und 999 aus.

Aufgabe 6 (15 Punkte) Schreiben Sie eine Funktion, die

- a) das größte Element einer Liste zurückliefert.
- b) ohne eine neue Liste zu erzeugen eine Liste umdreht.
- c) testet, ob ein Element `a` in einer Liste vorkommt.
- d) eine Liste als Argument bekommt und eine neue Liste mit den Elementen an den ungeraden Positionen zurückliefert
- e) die laufende Summe einer Liste berechnet.

```
lauf_summe([1, 3, 4, -7, 5]) => [1, 4, 8, 1, 6]
```

Aufgabe 7 (6 Punkte) Schreiben Sie drei Funktionen, die alle die Summe der Zahlen in einer Liste ausrechnen. Zum einen mit Hilfe einer `for`-Schleife, dann mit Hilfe einer `while`-Schleife und zuletzt mit Hilfe von Rekursion.

Aufgabe 8 (9 Punkte) Schreiben Sie eine Funktion `auf_aller` die eine Funktion und eine Liste als Argumente bekommt und eine neue Liste liefert, die die Rückgabewerte der Funktion auf jedes Listenelement enthält:

```
auf_aller(f, [1, 2, 3]) => [f(1), f(2), f(3)]
```

Verwenden Sie `auf_aller` um eine Funktion zu schreiben, die die Liste der Quadratzahlen von 1 bis n^2 zurückgibt. Falls es Ihnen nicht gelungen ist, `auf_aller` zu schreiben, verwenden sie die eingebaute Funktion `map`.