Übung 1 Abgabe 8.5.2020

In dieser Übung sollen Sie das algorithmische Denken üben (Aufgaben 1 und 2) sowie Grundlagen der Programmiersprache Python üben bzw. wiederholen (Aufgabe 3).

Aufgabe 1 – Freitag der 13. 8 Punkte

Alle paar Monate erleben wir Freitag, den 13., aber wie oft kommt dieses Datum eigentlich vor?

a) Wie viele verschiedene Fälle gibt es, wie die Tage eines Jahres auf die Wochentage fallen können? Denken Sie dabei auch an Schaltjahre! Bestimmen Sie für jeden dieser Fälle, wie oft Freitag der 13. in dem betreffenden Jahr vorkommt.

b) Bestimmen Sie, wie oft Sie seit Ihrer Geburt schon Freitag, den 13. erlebt haben.

Aufgabe 2 – Stabile Gläser 16 Punkte

Gläser gehen in Gaststätten häufiger zu Bruch. Deshalb werden ständig verbesserte Glassorten entwickelt. Um die Stabilität einer neuen Sorte zu überprüfen, wird folgendes Experiment durchgeführt: Es steht ein 3 m hoher Ständer zur Verfügung, an dem im Abstand von 10 cm Plattformen angebracht sind, von denen man ein Glas kontrolliert fallen lassen kann. Es soll mit möglichst wenigen Versuchen die "Fallhöhe" der Glassorte bestimmt werden, d.h. die größte Höhe, bei der ein fallendes Glas unbeschädigt bleibt. Als Versuch gilt dabei der Sturz von einer Plattform. Allerdings ist die Anzahl der Gläser, die beim Experimentieren zu Bruch gehen dürfen, beschränkt – das macht die Aufgabe kompliziert.

a) Wie bestimmt man die maximale Fallhöhe, wenn man nur ein einziges Glas zur Verfügung hat? Wie viele Versuche werden maximal benötigt (mit Begründung)? Aus Sicht des algorithmischen Aufwands ist dies der "ungünstige Fall" – für die Gaststätten ist er hingegen erwünscht.

b) Man kann in weniger Versuchen zum Ziel kommen, wenn man bereit ist, mehr Gläser zu opfern. Wie geht man vor, wenn die Zahl der zerbrochenen Gläser nicht beschränkt ist? Wie viele Versuche benötigt man jetzt maximal, und wie viele Gläser sind danach höchstens zerbrochen?

c) Ein Kompromiss ergibt sich, wenn zwei Gläser zu Bruch gehen dürfen. Welches Vorgehen minimiert nun die Anzahl der Versuche im "ungünstigen Fall", und wie viele Versuche werden dafür benötigt? Tipp: 15 oder 16 Versuche sind nicht optimal, das geht noch besser.

d) Verallgemeinern Sie das Ergebnis aus c) für den Fall, dass der Ständer n Plattformen hat. Das Vorgehen bleibt dabei prinzipiell gleich, aber die Anzahl der Versuche im "ungünstigen Fall" ist jetzt eine Funktion $f(n)$. Bestimmen Sie diese Funktion! Da die Anzahl der Versuche eine ganze Zahl sein muss, hat $f(n)$ die Form einer Treppe: für einen gewissen Bereich von n bleibt $f(n)$ konstant, springt dann um eins nach oben, ist wieder konstant usw. Für n=30 sollte Ihr Ergebnis aus c) herauskommen.

Aufgabe 3 – Einführung in Python 16 Punkte

a) Installieren Sie die Programmiersprache Python auf Ihrem System (siehe www.python.org). Wenn Sie Linux verwenden, ist Python wahrscheinlich bereits installiert.

b) Arbeiten Sie aus dem Python-Tutorium (siehe docs.python.org/tutorial/) die Abschnitte 3 bis 5 sowie 7.2.1, 8 und 9.3 durch. Sie sollen dabei die angegebenen Beispiele auf Ihrer Maschine nachvollziehen. Die folgenden Unterabschnitte, die sich an fortgeschrittene Benutzer richten, können übersprungen werden: 4.7, 5.1.3, 5.1.4, 5.6 - 5.8. Verschaffen Sie sich außerdem unter docs.python.org/library/index.html einen Überblick darüber, welche vordefinierten Module Python bereits anbietet.

c) Beantworten Sie folgende Verständnisfragen (wo nötig mit Quellcode):

- In welchem Modul befindet sich die Funktion `sqrt()` für die Quadratwurzel, und wie importiert und verwendet man diese Funktion?
- Was passiert, wenn man `sqrt()` mit einer negativen Zahl aufruft?
- Implementieren Sie eine Funktion `mysqrt()`, die stattdessen die Ausgabe „`mysqrt()` funktioniert nicht für negative Zahlen, du Dussel!“ anzeigt (zwei Antworten: benutzen Sie `if: ... else:` sowie `try: ... except:`)!
- Schreiben Sie eine Schleife, die die Variable `i` von -10 bis +10 (inklusive) inkrementiert, und geben Sie bei jeder Iteration den Wert `i` sowie das Ergebnis von `i % 5` („i modulo 5“) aus. Erklären Sie die Ausgabe (d.h. die Funktionsweise des Modulo-Operators).
- Wann sollte man einen String in dreifache Anführungszeichen (`'''`) einschließen?
- Worin besteht der Unterschied zwischen der Klasse `list` und der Klasse `dict`?
- Wozu dient die `__init__()`-Funktion einer Klasse und wie benutzt man sie?

d) Implementieren Sie das Sieb des Eratosthenes zur Bestimmung von Primzahlen (den Algorithmus finden Sie im Internet). Geben Sie Ihre Lösung in einem File „`sieve.py`“ ab, das wie folgt benutzt werden kann:

```
>>> with open("sieve.py") as fp:  # open file
...     exec(fp.read())           # load function sieve()
>>> primes = sieve(1000)         # return array of all primes below 1000
>>> print(primes)                # print result
```

Bitte laden Sie Ihre Lösung bis zum 8.5.2020 um 12:00 Uhr auf Moodle hoch.