Universität des Saarlandes
FR 6.2 Informatik

Kurt Mehlhorn und Corinna Coupette **WiSe 2021/22**

Übungen zu Ideen der Informatik

https://www.mpi-inf.mpg.de/departments/algorithms-complexity/teaching/winter21/ideen/

Blatt 3: Algorithmen und Programme **Abgabeschluss: 15.11.2021**

Aufgabe 1 Notenspiegel *(13 Punkte)*

Sie haben die Aufgabe, die Ergebnisse einer Klausur auszuwerten. Gehen Sie davon aus, dass die Klausurnoten ganze Zahlen zwischen 0 und 15 sind (einschließlich). Die n Klausurnoten haben Sie in einer Liste `einzelnoten` festgehalten. Dabei ist `einzelnoten[i]` die Note des $i$ten Teilnehmers, beginnend bei Teilnehmer 0. Gehen Sie auch allgemein davon aus, dass die Indexierung von Listen bei 0 beginnt, d.h. eine Liste der Länge n hat als ersten Index 0 und als letzten Index $n-1$.
Eine Liste von Einzelnoten für eine Klausur mit fünf Teilnehmern könnte z.B. so aussehen (mit Indexierung):

`einzelnoten` =	10	4	7	12	15
Index	0	1	2	3	4

a) Vervollständigen Sie das nachfolgende Pseudocode-Programm so, dass am Ende des Programms die Häufigkeit von Note i in `notenspiegel[i]` steht. (3 *Punkte*)

Algorithmus 1 : Ermittlung des Notenspiegels

einzelnoten ← Eingabe
notenspiegel ← Liste der Länge 16, alle Einträge mit 0 initialisiert

Lösung:
Siehe Algorithmus 2.

Algorithmus 2 : Ermittlung des Notenspiegels

einzelnoten ← Eingabe
notenspiegel ← Liste der Länge 16, alle Einträge mit 0 initialisiert
for position von 0 bis $n-1$ **do**
    note ← einzelnoten[position]
    notenspiegel[note] ← notenspiegel[note] + 1

b) Vervollständigen Sie das nachfolgende Pseudocode-Programm so, dass es für einen Notenspiegel und die Anzahl n der Noten als Eingabe den Notendurchschnitt ausdruckt. (*4 Punkte*)

Hilfestellung: Wenn es 5 Mal die Note 2 und 7 Mal die Note 4 gegeben hat, was ist dann der Notendurchschnitt?

Algorithmus 3 : Ermittlung des Notendurchschnitts

```
notenspiegel ← Eingabe
n ← Eingabe
```

Lösung:
Siehe Algorithmus 4.

Algorithmus 4 : Ermittlung des Notenspiegels

```
notenspiegel ← Eingabe
n ← Eingabe
notensumme ← 0
for note von 0 bis 15 do
|   notensumme ← notensumme + notenspiegel[note] · note
drucke notensumme / n
```

c) Angenommen, Sie kennen auch das Geschlecht $g \in \{m, w, d\}$ Ihrer Studierenden, das in einer Liste `geschlechter` festgehalten ist, die dieselbe Reihenfolge hat wie die Liste `einzelnoten`. Verändern Sie Ihr Programm aus Aufgabenteil a) so, dass es für jedes Geschlecht separat den Notenspiegel ermittelt. (*4 Punkte*)

Lösung:
Siehe Algorithmus 5. Entscheidend ist hier, dass mehrere Listen geführt werden und die Einzelnoten nach Geschlecht in die verschiedenen Listen hineinsortiert werden.

d) Angenommen, Sie ermitteln im Anschluss an Aufgabenteil c) den Notendurchschnitt für die einzelnen Geschlechter und stellen fest, dass der Notendurchschnitt für ein Geschlecht zwei Punkte unter dem Notendurchschnitt der anderen beiden Geschlechter liegt. Was können Sie daraus ableiten und was nicht? (*2 Punkte*)

Lösung:
Aus dem Unterschied allein lässt sich nichts weiter ableiten, insbesondere nicht, dass ein Geschlecht generell schlechter in Ihrem Fach ist als die anderen oder dass Sie in Ihrer Bewertung nach Geschlecht diskriminieren. Insbesondere könnte der Unterschied auch Zufall oder das Ergebnis anderer Variabler sein, die Sie nicht miterfassen. Um aus Ihrer Beobachtung weitere Schlussfolgerungen zu ziehen, brauchen Sie statistische Methoden (und unter Umständen auch mehr oder andere Daten).

Aufgabe 2 Programmverständnis *(10 Punkte)*

In dieser Aufgabe sollen Sie ein Programm verstehen, das in der Programmiersprache Python geschrieben ist. Sie werden sehen, dass das geht, ohne dass wir die Syntax und Semantik dieser Programmiersprache explizit eingeführt haben.

Betrachten Sie die folgenden neun Zeilen Code.

Algorithmus 5 : Ermittlung des Notenspiegels

```
einzelnoten ← Eingabe
geschlechter ← Eingabe
notenspiegel_m ← Liste der Länge 16, alle Einträge mit 0 initialisiert
notenspiegel_w ← Liste der Länge 16, alle Einträge mit 0 initialisiert
notenspiegel_d ← Liste der Länge 16, alle Einträge mit 0 initialisiert
for position von 0 bis n − 1 do
    note ← einzelnoten[position]
    geschlecht ← geschlechter[position]
    if geschlecht = m then
        notenspiegel_m[note] ← notenspiegel_m[note] + 1
    else
        if geschlecht = w then
            notenspiegel_w[note] ← notenspiegel_w[note] + 1
        else
            notenspiegel_d[note] ← notenspiegel_d[note] + 1
    notenspiegel[note] ← notenspiegel[note] + 1
```

```
1 elemente = [3, 6, 1, 2, 5]
2 x = 0
3 y = 1
4 for element in elemente:
5     if element % 2 == 0:
6         x = x + element
7     else:
8         y = y * element
9 print(x + y)
```

Beantworten Sie nun die folgenden Fragen (mit Begründung):

a) Was sind „`elemente`“, „`x`“ und „`y`“ in der Terminologie der Vorlesung? (*1 Punkt*)

Lösung:
„`elemente`“, „`x`“ und „`y`“ sind Variablen, d.h. Container für Werte. Das kann man z.B. daraus ableiten, dass mit ihnen im Verlauf des Programmes weitergearbeitet wird.

b) Was macht „=“ in der Terminologie der Vorlesung, und welchem Pseudocode-Zeichen entspricht dies? (*1 Punkt*)

Lösung:
„=“ bewirkt eine Wertzuweisung, die in unserem Pseudocode mit ← dargestellt wird.

c) Wie häufig wird die `for`-Schleife in Zeile 4 ausgeführt? (*1 Punkt*)

Lösung:
Fünf Mal, da die Liste fünf Elemente hat.

d) Vor Ausführung der letzten Zeile hat `x` den Wert 8 und `y` den Wert 15. Was macht das Programm in der letzten Zeile? (*1 Punkt*)

Lösung:
Das Programm druckt die Summe von `x` und `y`, d.h. in diesem Fall 23.

e) Für zwei positive Zahlen x und y ist (x `%` y) der Rest der Ganzzahldivision von x durch y; z.B. ist 6 `%` 5 = 1 und 14 `%` 4 = 2.

Für welche Zahlen aus „`elemente`“ gilt „`element % 2 == 0`“?

Und allgemeiner: Welche Eigenschaft einer positiven Zahl prüft „`element % 2 == 0`“? (*2 Punkte*)

Hilfestellung: Falls Sie sich fragen, wo der Unterschied zwischen „=“ und „==“ im Python-Code liegt, schauen Sie ans Ende des aktuellen Foliensatzes.

Lösung:
Für 6 und 2, d.h. „`element % 2 == 0`“ prüft, ob eine Zahl gerade (also ohne Rest durch Zwei teilbar) ist.

f) Wie lautet die Ausgabe in der letzten Zeile, wenn Sie die erste Zeile des Programms durch „`elemente = [10,20,30,40]`“ ersetzen? (*1 Punkt*)

Lösung:
101. Da alle Zahlen gerade sind, hat `x` den Wert 100 und `y` den Wert 1. 101 ist die Summe von `x` und `y`.

g) Sei nun „`elemente`“ eine beliebige Liste natürlicher Zahlen. Welchen Wert hat `x` vor Ausführung der letzten Zeile, und welchen Wert hat `y` – abstrakt formuliert? (*2 Punkte*)

Lösung:
`x` ist die Summe aller geraden Zahlen in der Liste und `y` ist das Produkt aller ungeraden Zahlen in der Liste.

h) Was ist der kleinste Wert, den das Programm in der letzten Zeile ausgeben kann, wenn „`elemente`“ eine beliebige Liste natürlicher Zahlen ist? (*1 Punkt*)

Lösung:
Der kleinste denkbare Wert ist 1, z.B. dann, wenn die Liste leer ist oder nur die 1 enthält. Ein kleinerer Wert ist bei natürlichen Zahlen (unabhängig davon, ob man die Null dazu zählt) nicht möglich, denn die Summe gerader natürlicher Zahlen ist nie negativ und das Produkt ungerader natürlicher Zahlen ist mindestens Eins.

Aufgabe 3 Hexadezimalzahlen *(7 Punkte)*

Wir haben bisher das Dezimalsystem (Basis: 10) und das Binärsystem (Basis: 2) kennengelernt. Ein weiteres praktisch relevantes Zahlensystem ist das Hexadezimalsystem (Basis: 16) mit den Hexadezimalziffern 0-9 und A-F. Dabei steht A für die Ziffer 10 im Dezimalsystem, B für die Ziffer 11, usw. So entspricht z.B. die Zahl 200 im Dezimalsystem der Zahl C8 im Hexadezimalsystem, denn $200 = 12 \cdot 16^1 + 8 \cdot 16^0$.

a) Welcher Hexadezimalzahl entspricht die Dezimalzahl 123 (mit Herleitung)? (*1 Punkt*)

Lösung:
$123_{10} = 7 \cdot 16^1 + 11 \cdot 16^0 = 7\text{B}_{16}$.

b) Welcher Dezimalzahl entspricht die Hexadezimalzahl A4F (mit Herleitung)? (*1 Punkt*)

Lösung:
$\text{A4F}_{16} = 10 \cdot 16^2 + 4 \cdot 16^1 + 15 \cdot 16^0 = 2639_{10}$.

c) Wie vielen Bits entspricht eine Hexadezimalziffer (mit Begründung)? (*1 Punkt*)

Lösung:
$\log_2 16 = 4$ Bits.

d) Sie möchten elektronische Geräte mit *Unique Identifiers* (UIDs) versehen. Wie viele Geräte könnten Sie ausstatten, wenn die UIDs aus zwölf Hexadezimalziffern bestünden (mit Begründung)? (*2 Punkte*)

Lösung:
$16^{12} = 2^{4 \cdot 12} = 2^{48}$. So zu finden z.B. bei MAC-Adressen.

e) Nennen Sie zwei alltagsrelevante Anwendungen, in denen Hexadezimalzahlen verwendet werden. (*2 Punkte*)

Lösung:
Drei klassische Beispiele sind:

- Darstellung von Farben als Tripel mit Rot-, Grün- und Blau-Anteil im Web, z.B. #AABBCC.
- Mac-Adressen auf Hardware, z.B. auf dem Schild unter Ihrem Router, z.B. 01-23-45-67-89-AB.
- IPv6-Adressen, z.B. 2001:0db8:0000:0000:0000:ff00:0042:8329.

Aufgabe 4 Seminarorganisation *(10+5 Punkte)*

In dieser Aufgabe lernen Sie Zählen (Sie dachten, das können Sie schon? Umso besser!). Alle Aufgaben können Sie allein durch Nachdenken lösen. Wir setzen keine Vorkenntnisse voraus und die Reihenfolge der Teilaufgaben ist bewusst gewählt.

Sie organisieren ein Seminar mit 8 Teilnehmern.

a) Sie stellen eine Frage und niemand meldet sich. Wie viele Möglichkeiten haben Sie, einen Teilnehmer einfach so dranzunehmen? (*1 Punkt*)

   Lösung:
   8.

b) Sie stellen eine weitere Frage und es meldet sich wieder niemand. Wie viele Möglichkeiten haben Sie, einen Teilnehmer einfach so dranzunehmen, wenn Sie den Teilnehmer aus der vorigen Teilaufgabe nicht wieder drannehmen wollen? (*1 Punkt*)

   Lösung:
   $8 - 1 = 7$.

c) Jeder Teilnehmer muss genau ein Referat halten. In wie vielen verschiedenen Reihenfolgen können die 8 Teilnehmer sprechen? (*2 Punkte*)

   Lösung:
   Wir wissen bereits, wie viele Möglichkeiten es gibt, das erste Referat auszuwählen: 8. Dann gibt es 7 Möglichkeiten, das zweite Referat auszuwählen, usw. Die entscheidende Frage ist, wie sich diese Möglichkeiten zueinander verhalten (also: $8 + 7 + ...$, $8 \cdot 7 \cdot ...$, oder doch etwas ganz anderes?). Sie beobachten, dass es *für jedes mögliche* erste Referat (von denen es 8 gibt) 7 mögliche zweite Referate gibt (malen Sie sich einen Baum dazu auf). Das bedeutet:

$$8 \cdot 7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 8!$$

d) Angenommen, nur 5 Teilnehmer sollen Referate halten. Die Referate sind durchnummeriert von 1 bis 5. Wie viele Möglichkeiten gibt es, den Referaten Sprecher zuzuordnen? (*3 Punkte*)

   Lösung:
   Sie haben 8 Möglichkeiten, den ersten Referenten auszuwählen, 7 Möglichkeiten, den zweiten Referenten auszuwählen, usw. Insgesamt wählen Sie fünf Referenten aus. Daher:

$$8 \cdot 7 \cdot 6 \cdot 5 \cdot 4 = \frac{8!}{3!}$$

e) Wie viele Möglichkeiten gibt es, ohne Rücksicht auf die Referatsreihenfolge aus 8 Seminarteilnehmern 5 Sprecher auszuwählen? (*3 Punkte*)

   Lösung:
   Sie wissen bereits aus der vorigen Teilaufgabe, wie viele Möglichkeiten es gibt, den Teilnehmern Referate zuzuordnen, wenn die Reihenfolge eine Rolle spielt: $(8 \cdot 7 \cdot 6 \cdot 5 \cdot 4)$. Wir wollen nun alle Reihenfolgen gleich behandeln, die dieselben Teilnehmer beinhalten. Für jede Menge von fünf Teilnehmern gibt es $(5 \cdot 4 \cdot 3 \cdot 2 \cdot 1)$ mögliche Referatreihenfolgen. Um nur die verschiedenen Teilnehmermengen, nicht aber Reihenfolgen zu zählen, teilen wir also das Ergebnis der vorigen Teilaufgabe durch die möglichen Reihenfolgen. Damit gelangen wir zu:

$$(8 \cdot 7 \cdot 6 \cdot 5 \cdot 4)/(5 \cdot 4 \cdot 3 \cdot 2 \cdot 1) = \frac{8!}{5!3!} = \binom{8}{5}$$

f) (Bonus) Wie lauten die Antworten zu den vorigen Teilaufgaben für ein Seminar mit n Teilnehmern, von denen in den letzten beiden Teilaufgaben k Referate halten sollen? (*5 Punkte*)

Lösung:
Von oben nach unten:

(1) n

(2) $n - 1$

(3) $n \cdot ... \cdot 1 = n!$

(4) $n \cdot ... \cdot (n - k + 1) = \binom{n}{k} k!$

(5) $\big(n \cdot ... \cdot (n - k + 1)\big) / \big(k \cdot ... \cdot 1\big) = \binom{n}{k}$

$n!$, gesprochen „n Fakultät“, ist nichts weiter als eine Abkürzung für $n \cdot ... \cdot 1$, also die Anzahl von Möglichkeiten, n unterscheidbare Elemente in eine Reihenfolge zu bringen (man nennt diese Reihenfolgen auch *Permutationen*).

Ebenso ist $\binom{n}{k}$, gesprochen „n über k“ oder (besser) „k aus n“ (engl. „n choose k“), nichts weiter als eine Abkürzung für $\big(n \cdot ... \cdot (n-k+1)\big) / \big(k \cdot ... \cdot 1\big)$, also die Anzahl von Möglichkeiten, k aus n unterscheidbaren Elementen auszuwählen, wenn Wiederholungen nicht erlaubt sind und die Reihenfolge keine Rolle spielt.

Ich habe für die Videos, die Nachbereitung und das Übungsblatt etwa ☐ Stunden gebraucht.
(Ann-Sophie fertigt aus diesen Zahlen eine Statistik an. Kurt und Corinna sehen nur diese Statistik. Wir möchten wissen, ob der Schwierigkeitsgrad in etwa richtig ist.)

Algorithmen und Programme war spannend ☐ okay ☐ langweilig ☐
schwierig ☐ okay ☐ einfach ☐