

9. Theelachter MaPhIA-Rundbrief (April 2021)

Dieser Brief behandelt wieder **Mathematik-**, **Physik-** und **Informatik-Angelegenheiten**. Alle Briefe und Lösungen sind unter www.ymmij.de/MaPhIA/ nachzulesen. Fragen und Anregungen auch gerne unter info@ymmij.de.

Warnung: Bei Mathe-Phobie ganz vorsichtig weiterlesen!

Viel Spaß!

Jimmy Brüggemann

Problem 1: Ein unfaires Spiel

Spielregel: Du setzt 1 Euro auf eine der sechs Zahlen eines Würfels. Dann werden drei Würfel gleichzeitig geworfen.

- ▶ Ist die von Dir gewählte Zahl nicht dabei, verlierst Du Deinen Einsatz (1 Euro).
- ▶ Sonst bekommst Du Deinen Einsatz zurück und für jeden Würfel, der Deine Zahl zeigt, einen weiteren Euro. Du kannst also bis zu 3 Euro gewinnen.

Warum ist das Spiel unfair? Kannst Du dieses Spiel fair machen?

Du willst das Spiel durch einen hohen Gewinn attraktiv machen. Was kannst Du ändern?

P.S.: Bei einem fairen Spiel machen alle Beteiligten auf lange Sicht (wenn der Zufall ausgeglichen ist) weder Gewinn noch Verlust.

Problem 2: Elf Teiler

Die kleinste Zahl mit 10 Teilern ist offenbar 48, die kleinste Zahl mit 12 Teilern ist 60. Wie heißt die kleinste Zahl, die genau 11 Teiler hat?

Problem 3: Farbenspiele

Chamäleons wechseln ihre Farbe, um mit ihren Artgenossen zu kommunizieren.

In einem Terrarium leben 7 spezielle Chamäleons. Jedes Tier kann drei Farben annehmen: rot, blau oder grün. Zur Zeit sind vier von ihnen rot gefärbt, zwei blau und eins ist grün.

Meist gehen sich Chamäleons aus dem Weg. Begegnen sich doch zwei Tiere, wechseln beide die Farbe – aber nur, wenn beide unterschiedlich gefärbt sind. Beide nehmen dann die dritte Farbe an. Treffen sich z.B. ein blaues und ein grünes Tier, sind danach beide rot.

Können irgendwann alle Chamäleons dieselbe Farbe haben? Falls ja, welche?

Wie ist die Situation, wenn Du zwei rote Chamäleons dazubekommst?

Problem 4: Pflastern!

Neele hat zwei unterschiedlich große, rechteckige Terrassen mit quadratischen Gehwegplatten gepflastert. Jede Terrasse ist einreihig mit roten Platten umrahmt; innen liegen graue Platten. Beide Terrassen enthalten genauso viele rote wie graue Platten.

Wie viele Platten hat Neele verlegt?

----- ab in den Briefkasten von Theelacht 7 -----

Name:

Ich will die Lösungen wissen.

Adresse:

Ich habe sogar eine Aufgabe gelöst!

E-Mail:

Ich will den Rundbrief per E-Mail.

Alter: 0-9 10-14 15-24 25-59 60+

Verschone mich bloß mit MaPhIA !!!

Denken wie ein Informatiker

Ausgangssituation: Auf einem Bauernhof liegen zwölf schwere Baumscheiben verschiedener Größe gestapelt neben dem Haus. Dort stören sie nun und sie sollen neben den Stall gebracht werden. Der Traktor kann leider nur eine Scheibe auf einmal bewegen. Die Berufsgenossenschaft verlangt zur Unfallverhütung, dass nie eine Scheibe größeren Durchmessers auf einer kleineren liegt. Daher sind die Scheiben am Haus der Größe nach gestapelt. Der Hof ist recht voll, es gibt nur einen Platz, wo zwischendurch ein weiterer Stapel von Baumscheiben entstehen kann. Für jede Baumscheibe stehen folgende Daten zur Verfügung: Baumart, Durchmesser, Dicke, Volumen, Gewicht, Marktwert.

Problem: Welche Daten werden zur Lösung benötigt, welche sind überflüssig, welche fehlen?

Nachdem wir uns vergewissert haben, dass der Traktor auch die schwerste Scheibe bewegen kann, bleibt folgendes Szenario: Es gibt drei Plätze (Haus, Stall, Hof) und 12 unterschiedlich große Scheiben, die wir der Größe nach durchnummerieren (die kleinste mit 1). Zu keinem Zeitpunkt darf eine größere Scheibe auf einer kleineren liegen. Zu Beginn liegt der nach Scheibengröße geordnete Turm am Haus, am Ende soll er –genauso geordnet– am Stall stehen.

Abstraktion: Kann man das Problem in –voneinander unabhängige– Teilprobleme zerlegen oder auf einfachere Fälle des gleichen Problems zurückführen?

Können wir das Problem allgemeiner lösen und die Lösung dann wiederverwenden, falls in einem anderen Fall z.B. 14 Scheiben zu bewegen sind oder mehrere Zwischenlager zu Verfügung stehen?

Wir wollen das Problem für eine beliebige Anzahl von Scheiben und einem Zwischenlager lösen.

Rekursion: Wir führen das Problem für n Scheiben auf dasselbe Problem mit $n - 1$ Scheiben zurück:

Die Verlagerung eines Turms der Höhe n von A nach B mit Hilfe eines Platzes C geschieht, indem

1. ein Turm der Höhe $n - 1$ von A nach C (mit Hilfe von B) verlagert wird,
2. die größte Scheibe (mit der Nummer n) von A nach B bewegt wird,
3. der Turm der Höhe $n - 1$ von C nach B (mit Hilfe von A) verlagert wird.

Damit ist das Problem auf zwei einfachere (!) Teilprobleme und eine Traktorfahrt zurückgeführt.

Codierung: Ein erster Programmwurf sieht dann vielleicht so aus:

```
verlege_turm (12, "Haus", "Stall", "Hof") wobei
algorithmus verlege_turm (N, A, B, C) definiert ist durch
  verlege_turm (N-1, A, C, B)
  schreibe ("Traktor fährt Scheibe ", N, " von ", A, " nach ", B, ".\n")
  verlege_turm (N-1, C, B, A)
```

Terminierung: Leider funktioniert das Programm nicht! Für Menschen ist klar, dass man fertig ist, wenn ein Turm leer ist, aber der Computer rechnet unbeirrt mit negativen Zahlen weiter!

Wir ergänzen das Programm, so dass nur etwas passiert, solange ein Stapel noch Scheiben enthält:

```
verlege_turm (12, "Haus", "Stall", "Hof") wobei
algorithmus verlege_turm (N, A, B, C) definiert ist durch
falls N > 0 dann
  verlege_turm (N-1, A, C, B)
  schreibe ("Traktor fährt Scheibe ", N, " von ", A, " nach ", B, ".\n")
  verlege_turm (N-1, C, B, A)
```

Analyse des Algorithmus: Das Programm liefert eine Lösung mit Anweisungen für den Traktor.

Gibt es weitere Lösungen? Ja, aber leider nur schlechtere. Wie oft muss der Traktor jetzt fahren?

Oh, für n Scheiben offenbar $2^n - 1$ -mal. Für 12 Scheiben also 4095-mal.

Konsequenz: Der Bauer räumt den Hof auf und schafft Abstellplätze für weitere Zwischenstapel (bei 11 Plätzen auf dem Hof wird die Sache sehr einfach, bei 6 Plätzen auch nicht schwer, schon ein zweites Zwischenlager reduziert die Anzahl der Fahrten auf 81, ein drittes auf 47).

Der Algorithmus in der Programmiersprache Haskell (Aufruf: `verlege_turm 12 "Haus" "Stall" "Hof"`):
`verlege_turm 0 = []`
`verlege_turm N A B C = verlege_turm (N-1) A C B ++ [(N,A,B)] ++ verlege_turm (N-1) C B A`