

37. Bundeswettbewerb Informatik

Anregungen für den Unterricht



Johannes Pieper, Bundeswettbewerb Informatik Alumni und Freunde e. V.

31. August 2018

Liebe Lehrerinnen und Lehrer,

wir möchten Ihnen ein paar Hinweise an die Hand geben, wie Ihren Schülerinnen und Schülern der Einstieg in die Aufgaben der ersten Runde des 37. Bundeswettbewerb Informatik erleichtert werden kann. Mit ihnen lassen sich fast alle Aufgaben erfahrbar machen. Auf diese Weise ist es möglich, einen ersten Eindruck der Problemstellung zu gewinnen und auch die ersten Ansätze für eine mögliche Lösungsstrategie zu erarbeiten. Auch ein paar Hinweise bzgl. der Programmierung sind enthalten.

Dazu ein allgemeiner Hinweis aus Einsendungen der letzten Jahre, der weitergegeben werden sollte: Nicht jede gerade vorher im Unterricht gelernte Datenstruktur ist damit für die Lösung der Aufgaben geeignet (wenn man einen Hammer hat, kommt es einem so vor, dass man nur noch Nägel sieht). Das selbe gilt auch für Algorithmen.

Natürlich können Sie diese Hinweise auch direkt an die Schülerinnen und Schüler weitergeben. Deshalb sind diese Hinweise auch entsprechend formuliert. Die Erfahrung zeigt, dass durch eine Behandlung der Problemstellung im Unterricht mehr Schülerinnen und Schüler am Wettbewerb teilnehmen, da die Einstiegshürden gesenkt werden.

Die Aufgaben zur ersten Runde finden Sie unter der Adresse

<https://bwinf.de/bundeswettbewerb/37-bwinf/1-runde/>

auf den Seiten von *BWINF: Bundesweit Informatiknachwuchs fördern*.

Allgemeines

Da bei mehreren Aufgaben die Eingaben aus Dateien eingelesen werden sollen, ist es zu empfehlen, das Einlesen von Textdateien zu können. Im Anhang sind ein paar Code-Schnipsel zu sehen, mit denen eine Datei für die Juniaraufgabe 2 eingelesen werden kann. Dabei werden die Daten aber nicht alle intern gespeichert.

Erfahrungsgemäß ist die Umsetzung einer informalen Lösungsidee in formale Algorithmen und Datenstrukturen und dann auch in ein Programm gerade für neue Teilnehmerinnen und Teilnehmer schwer. Leider kann in diesem Dokument dazu nicht viel gesagt werden, ohne genauere Lösungsideen zu verraten. Als Lehrkräfte können Sie aber bei konkreten Fragen Ihrer Schülerinnen und Schüler entsprechende Hinweise und Anmerkungen geben.

Bundeswettbewerb Informatik Alumni & Freunde e.V. · c/o Robert Czechowski · Richard-Wagner-Str. 59 · 53115 Bonn
WWW: <http://alumni.bwinf.de> · E-Mail: vorstand@alumni.bwinf.de

Ansprechpartner für diesen Text: Johannes Pieper · E-Mail: infounterricht@yahoo.de

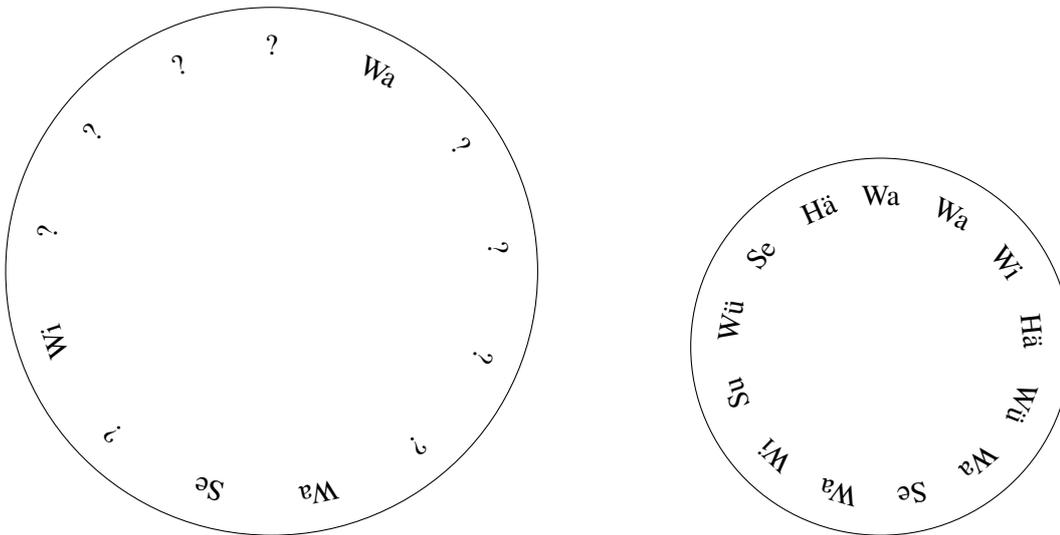
Auf und Ab (Junioraufgabe 1)

Diese Aufgabe kann zuerst von Hand ausprobiert werden. Dazu kann das Spielfeld von der Webseite genommen werden. Beim Ausprobieren kannst Du merken, welche Informationen von dem Spielfeld für die Aufgabe relevant und welche nicht relevant sind.

Bei der Umsetzung in ein Programm musst Du dann schauen, wie Du erkennen kannst, ob eventuell das Ziel nicht erreicht werden kann. Dazu kannst Du am Spielfeld ausprobieren, wann man erkennen kann, dass das Ziel nicht erreicht wird.

Baywatch (Junioraufgabe 2)

Auch bei diese Aufgabe kann man sehr gut einfach mal von Hand ausprobieren. Das Beispiel aus der Aufgabenstellung wurde dazu hier auf zwei verschiedene Arten aufbereitet. Einmal als Drehscheiben und einmal in Streifenform, wobei eine der Listen nur aus praktischen Gründen zweimal hintereinander geschrieben wurde. Schneide die kleine Schreibe und den kleinen Streifen aus und nutze Sie an dem zugehörigen Gegenstück. So solltest Du ein Gefühl dafür bekommen, was Du in einem passenden Programm machen musst.



Wa Wa Wi Hä Wü Wa Se Wa Wi Su Wü Se Hä	Wa Wa Wi Hä Wü Wa Se Wa Wi Su Wü Se Hä
--	--

? Wa ? ? ? ? Wa Se ? Wi ? ? ?

Superstar (Aufgabe 1)

Bei dieser Aufgabe sollte man sich zuerst eine Gruppe und wer darin wem folgt aufzeichnen. Hier ist die Empfehlung dieses in Form einer Tabelle bzw. Matrix zu machen, in der Du die Mitglieder der Gruppe mit ihren Beziehungen einträgst. Mit dieser Tabelle kannst eine Strategie entwickeln, welche Anfragen wirklich nötig sind und welche Anfragen Du schon durch das Ergebnis anderer Anfragen ausschließen kannst.

Twist (Aufgabe 2)

Für die Teilaufgabe 1 musst Du Dir überlegen, wie Du einzelne Worte in einem Text so umänderst, dass die Worte getwistet sind. Dazu kannst Du Dir zu mehreren Worten mögliche Twist-Varianten ausdenken. Darauf aufbauend kannst Du einen Twist-Algorithmus vorschlagen. Diesen kannst Du wieder ohne Computer ausprobieren und dann in ein Programm umsetzen.

Für die Teilaufgabe 2 ist die Frage, welches Wort aus der Wörterliste zu dem getwisteten Wort passt. Dazu kannst Du Dir überlegen, welche Eigenschaften ein Twist-Wort und ein Original-Wort haben müssen, damit sie zueinander passen. Dies kannst Du wieder an einer Liste von möglichen Twist-Worten zu einem Wort ausprobieren. Dazu kannst Du auch schauen, woran man erkennen kann, dass zwei Wörter nicht zueinander passen. Die Prüfung dieser Eigenschaften lässt sich dann in einem Programm umsetzen.

Voll daneben (Aufgabe 3)

Fange bei dieser Aufgabe klein an: Ein Zahlenbereich 1 bis 100 reicht für das arbeiten per Hand vollkommen aus. Setze nur für wenige Leute eine Zahl. Auch Al Capone darf dann nur wenige Zahlen auswählen. Fange dabei mit einer Zahl an und überlege, wo Du diese platzieren würdest. Wähle dann eine Platzierung für zwei, drei usw. Zahlen. Berechne dabei immer was Al Capone auszahlen muss und schaue Dir auch an, was passiert, wenn Du deine Wahl etwas verschiebst.

Insgesamt musst Du dabei gute Kriterien finden, wie die Zahlen platziert werden. Erarbeite diese so, dass ein Computerprogramm diese auch direkt berechnen kann.

Schrebergärten (Aufgabe 4)

Dieses ist eine klassische Aufgabe um erst einmal per Hand tätig zu werden: Passende Stücke aus Papier ausschneiden und selbst versuchen ein gutes Ergebnis zu erreichen. Zeichnungen für die ersten beide Beispiele in unterschiedlichen Maßstäben sind mitgeliefert.

Der schwierigste Teil der Aufgabe besteht darin, die so gefundene Strategie in ein Programm umzuwandeln.

1: 25 x 15

2: 6 x 3

1: 30 x 15

2: 2 x 2

2: 3 x 1

1: 25 x 20

2: 4 x 4

1: 25 x 15

2: 4 x 4

Widerstand (Aufgabe 5)

Hier kannst Du Dir zuerst aufschreiben, wie die (bis zu vier) Widerstände zu einem neuen Widerstand kombiniert werden können. Auf diese Kombinationen kannst Du dann die Widerstände verteilen, um zu schauen, ob das Ergebnis nahe des benötigten Widerstandes ist.

Mache Dir dann Gedanken, wie die Kombinationen entstanden sind und mittels eines Programms erzeugt werden können, um darauf eine Suche zu starten, die die ideale Kombination findet.

Einlesen von Dateien

In diesem Abschnitt wird gezeigt, wie die Beispieldateien aus der Junioraufgabe 2 eingelesen werden können. Jede Beispieldatei enthält:

- in der ersten Zeile die von George gefundenen Landzungen, im Uhrzeigersinn, getrennt durch Leerzeichen und ohne Fragezeichen.
- und in der zweiten Zeile die Landzungen von der Karte, im Uhrzeigersinn, beginnend im Norden, getrennt durch Leerzeichen und evtl. mit Fragezeichen

Die Code-Beispiele lesen nur die Daten aus der Datei aus und speichern nur die Daten, die für das gesamte Einlesen notwendig sind. Die Code-Beispiele lesen eine Datei ein und geben die Länge der beiden Listen aus. Deshalb können sie nicht ohne ein paar Ergänzungen für die konkrete Bearbeitung der Aufgabe genutzt werden und müssen noch etwas angepasst werden.

Java

Diese Methode wurde mit Java 8 erstellt.

```
1 private static void leseDatei(String dateiName) {
2     try (BufferedReader br = new BufferedReader( new FileReader(dateiName) )) {
3         // erste Zeile lesen
4         String georgeZeile = br.readLine();
5         String[] georgeListe = georgeZeile.split(" ");
6         int anzahlGeorge = georgeListe.length;
7         // zweite Zeile lesen
8         String kartenZeile = br.readLine();
9         String[] kartenListe = kartenZeile.split(" ");
10        int anzahlKarte = kartenListe.length;
11        System.out.println("Anzahl Landzungen George: " + anzahlGeorge);
12        System.out.println("Anzahl Landzungen Karte: " + anzahlKarte);
13    } catch (NumberFormatException | IOException e) {
14        System.out.println("Fehler beim einlesen der Datei " + dateiName);
15        e.printStackTrace();
16    }
17 }
```

Python

Dieser Python-Code wurde mit Python 3 erstellt.

```
1 fileName = "baywatch1.txt"
2 with open(fileName, "r") as f:
3     georgeZeile = f.readline()
4     georgeListe = georgeZeile.split()
5     kartenZeile = f.readline()
6     kartenListe = kartenZeile.split()
7     print ("Anzahl Landzungen George: " + str(len(georgeListe)))
8     print ("Anzahl Landzungen Karte: " + str(len(kartenListe)))
```