39. Bundeswettbewerb Informatik
Anregungen für den Unterricht

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

31. August 2020

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 39. 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/39/1/`

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 Code-Schnipsel zu sehen, mit denen eine Datei für die Junioraufgabe 2 eingelesen werden kann. Dabei werden die Daten aber nicht intern gespeichert, so dass die Code-Schnipsel nicht ohne kleine Anpassungen direkt verwendet werden können.

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: `bwinf@johpie.de`

Passwörter (Junioraufgabe 1)

Die Überlegungen, wie ein leicht zu merkendes Passwort aussehen kann, können direkt bei dieser Aufgabe verwendet werden. Eine Möglichkeit ist, dass man sich z. B. auf einer der verschiedenen Seiten dazu im Internet ein Passwort erstellen lässt und dann überlegt, welche Stellen an diesem Passwort es schwerer machen, es sich zu merken.

Bei der Umsetzung ist zu beachten, dass es drei eigene Regeln sein müssen, die im Programm verwendet werden. Die in der Aufgabenstellung angegebenen Regeln können, müssen aber nicht im Programm eingebaut werden.

Baulwürfe (Junioraufgabe 2)

Für die Erkennung von Baulwurfsbauen müssen die Hügel auf der gesamten Karte genauer analysiert wenden. Wenn ein einzelnes Feld mit einem Hügel betrachtet wird, was muss dann gemacht werden, um dieses Feld als Teil eines Baulwurfsbau zu identifizieren? Welche anderen Felder müssen alle dazu mit betrachtet werden? Bei dieser Analyse darf aber auch nicht vergessen werden, dass kein einziger Baulwurfsbau doppelt gezählt werden darf.

Ein mögliches Vorgehen kann an den folgenden einfachen Konstellationen überprüft werden.

Beispiel 1

```
     X X
XXX      X
X X  XXX
X X  X XXXX
XXX  X XX X
     XXXX X
 X X    XXX
```

Beispiel 2

```
X  XXX
   X X  X
XXXX X
X XXXX
X X    X
XXX  X   X
         X
```

Beispiel 3

```
XXXXXX
X XX X
X XX X
XXXXXX
```

Beispiel 4

```
X   X   X
  X   X   X
X
```

Wörter aufräumen (Aufgabe 1)

Ein mögliches Vorgehen ist es, sich zuerst selber entsprechende kleine Sätze auszudenken, die dann gelöst werden müssen. So kann man auf die Punkte kommen, die für eine Zuordnung der Wörter an die Plätze nötig sind. Im Programm muss dieses dann in die andere Richtung umgesetzt werden.

Zur Übung sind hier drei Beispiele angegeben, bei denen auch zu klären ist, ob sie eindeutig lösbar sind. Für die eigentliche Aufgabe ist jedes Rätsel aber eindeutig lösbar.

```
_e_ __n_ ____b_ __g__ ____ ___ _a__.
das Der Land Regen über treibt Wind

__s __s _n_ ___h___ ____t, __t _a_ ____t.
Das das lässt Licht ist wachsen was uns

_t____ _a____ _i__ _i__.
sind Stille tief Wasser
```

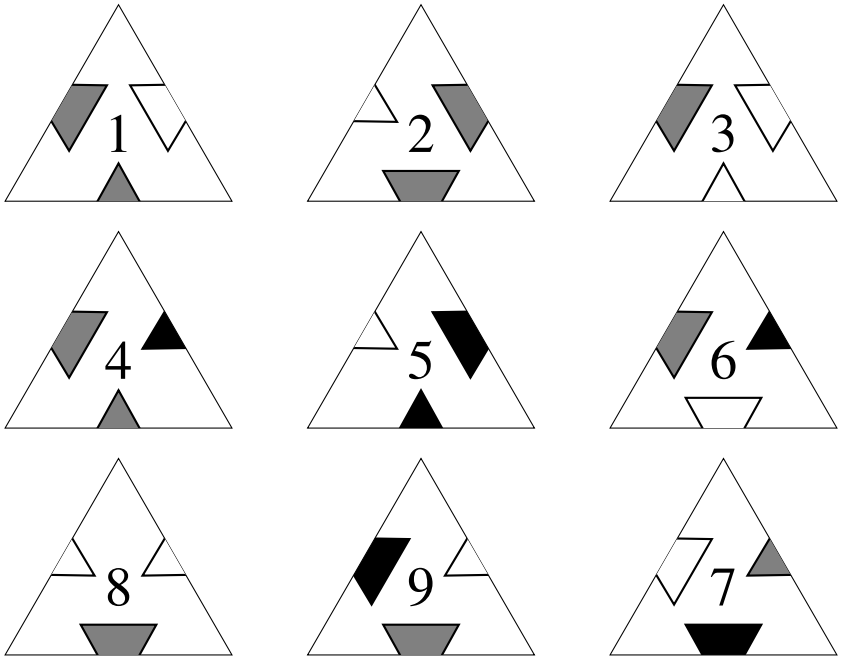
Dreieckspuzzle (Aufgabe 2)

Beim Dreieckspuzzle muss mit Hilfe eines planmäßigen Vorgehens herausgefunden werden, ob sich das Puzzle lösen lässt oder sicher nicht. Daher muss dieses systematische Vorgehen erarbeitet werden, wie und in welcher Reihenfolge zur

möglichen Lösung des Puzzle vorgegangen wird.

Eine Möglichkeit, die Vorgehensweise zu erarbeiten besteht darin, per Hand das Puzzle versuchen zu lösen und dabei immer die gleichen Schritte durchzuführen. Dieses kann auch nur mit einem Teil des Puzzles geschehen, wie einem Dreieck aus vier Teilen oder dem Kern aus sechs Elementen.

Um die Übersicht zu behalten, gibt es dafür Puzzleelemente, die neben Symbolen auch durchnummeriert sind. Hier sind die Symbole durch Dreiecke mit grauer, schwarzer und weißer Füllung ersetzt worden.



Tobis Turnier (Aufgabe 3)

Tobis Turnier lässt sich auch mit Zahlen zwischen 0 und 10 durchführen. In dieser Anzahl kann man auch kleine Plättchen in zwei verschiedenen Farben nutzen und damit verschiedene Turnierarten per Hand durchspielen. So bekommt man ein Gefühl für die verschiedenen Varianten.

Zwei Punkte sollten bei dieser Aufgabe beachtet werden: Nach welchen Kriterien wird die Empfehlung für Tobi ausgesprochen und wie viele Durchgänge sind nötig um eine Aussagekraft zu haben. zeileSpalten

Streichholzrätsel (Aufgabe 4)

Diese Aufgabe besteht aus zwei Teilen: Wie die Anordnungen im Computer gespeichert werden können, ist ein Problem bei dem bereits kleine Kenntnisse über Datenstrukturen sehr hilfreich sind.

Ob sich Muster in andere übertragen lassen, kann man dagegen auch mit Papier überprüfen und daraus eine Umsetzung als Programm ableiten. Dazu werden beiden Anordnungen auf Papier gemalt und gegen das Licht übereinander gelegt.

Wichteln (Aufgabe 5)

Beim Wichteln muss man eine Strategie finden, um eine möglichst gute Verteilung der Wichtelgeschenke zu erreichen. Ein Vorschlag zur Suche dieser Strategie ist, erste Situationen zu betrachten, die aus wenigen Schülern bestehen. Bei drei Schülern lassen sich schon sehr unterschiedliche Konfigurationen überlegen, für die dann die beste Verteilung per Hand gesucht werden kann.

Im nächsten Schritt sollte die Anzahl der Schüler erweitert werden, um die damit aufkommenden Probleme zu betrachten und zu verallgemeinern. Ein Vorgehen, bei dem einfach alle Möglichkeiten ausprobiert werden, sogenanntes Brute-Force, ist maximal bei der Betrachtung kleiner Schülergruppen zu nutzen, um bei der Strategiesuche zu helfen.

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 Breite der Karte.
- In der zweiten Zeile die Höhe der Karte.
- Es folgen die Reihen mit der Karte.

Die Code-Beispiele lesen nur die Daten aus der Datei aus und speichern nur die Daten, die für das gesamte Einlesen notwendig sind. Die Daten der Karten werden noch nicht gespeichert. Deshalb können diese Beispiele 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.

```java
public static void leseKarte(String dateiName) throws IOException {
  try (BufferedReader br = new BufferedReader(new FileReader(dateiName))) {
    // Lese Zeile mit der Breite
    String zeileBreite = br.readLine();
    int breite = Integer.parseInt(zeileBreite);
    // Lese Zeile mit Höhe
    String zeileHoehe = br.readLine();
    int hoehe = Integer.parseInt(zeileHoehe);
    for (int reihe = 1; reihe <= hoehe; reihe++) {
      // Lese die Zeile aus der Datei für diese Reihe
      String zeile = br.readLine();
      for (int spalte = 0; spalte < breite; spalte++) {
        // Dies sind nun die Werte der einzelnen Positionen.
        // Diese müssen irgendwie gespeichert werden.
        char quadrat = zeile.charAt(spalte);
      }
    }

  } catch (NumberFormatException | IOException e) {
    // Fehler, wenn die Datei nicht gelesen werden kann.
    System.out.println("Fehler beim Einlesen der Datei " + dateiName);
    throw e;
  }
}
```

Python

Dieser Python-Code wurde mit Python 3 erstellt.

```
fileName = "karte1.txt"
with open(fileName, "r") as f:
  # Lese Zeile mit Breite
  zeileBreite = f.readline()
  breite = int(zeileBreite)
  # Lese Zeile mit Höhe
  zeileHoehe = f.readline()
  hoehe = int(zeileHoehe)
  for reihe in range(hoehe):
    # Lese Zeile aus der Datei für diese Reihe
    zeile = f.readline()
    for spalte in range(breite):
      # Dies sind nun die Werte der einzelnen Positionen.
      # Diese müssen irgendwie gespeichert werden.
      quadrat = zeile[spalte]
```