

40. Bundeswettbewerb Informatik

Anregungen für den Unterricht



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

9. September 2021

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 40. 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 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/40/1/>

auf den Seiten der Bundesweiten Informatikwettbewerbe (BWINF).

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

Zum Winde verweht (Junioraufgabe 1)

Bei der Bearbeitung dieser Aufgabe muss die Entfernung zwischen einer Windkraftanlage und einem Wohngebäude berechnet werden. Eine Zeichnung einer Situation mit mehreren Wohngebäuden und einer Windkraftanlage könnte dabei hilfreich sein. Mit Hilfe einer solchen Zeichnung wäre es auch möglich sich überlegen, ob überhaupt für alle Kombinationen von Windkraftanlagen und Wohngebäuden die Entfernung berechnet werden muss. Zu beachten ist, dass bei der Umsetzung einer solche Überprüfung der zusätzliche Programmieraufwand im Verhältnis stehen sollte zum Beschleunigung der Berechnung.

Bereits bei der Bearbeitung eines Falles von Hand ist festzulegen, welche Elemente (Wohngebäude und Windkraftanlagen) in welcher Reihenfolge behandelt werden. Erweist sich diese als praktikabel, sollte sie auch bei der Programmierung übernommen werden. Deshalb müssen die Regeln für die Reihenfolge eindeutig formuliert sein.

Treffsicherheit (Junioraufgabe 2)

Bei dieser Aufgabe lohnt es sich ein paar Überlegungen im Vorfeld anzustellen. Dazu gehört, sich einen Weg zu überlegen, wie man einen Termin für eine einzelne Person am einfachsten, also mit möglichst wenig Änderungen, zum Beliebtesten macht. Auf der Grundlage dieser Überlegungen lassen sich auch einfache Aussagen über die insgesamt nötigen Änderungen treffen.

Schiebeparkplatz (Aufgabe 1)

Aussagen über bereits freie Autos zu treffen ist einfach. Bei blockierten Autos sieht dieses ganz anders aus. Besonders dann, wenn absehbar mehrere Fahrzeuge bewegt werden müssen. Verschiedene Situationen lassen sich mit 2x4 Klemmbausteinen (zum Beispiel von Lego) oder ausgeschnittenen Papierstücken, die doppelt so lange wie breit sind, nachstellen.

Für die Umsetzung in ein Programm muss überlegt werden, wie auf Grundlage der Daten erkannt wird, ob ein Auto frei ist oder nicht. Im zweiten Schritt muss dann überlegt werden, wie das Vorgehen aus der praktischen Umsetzung mit Papier oder Klemmbausteinen auf das vorgegebene Datenformat angewandt werden muss.

Vollgeladen (Aufgabe 2)

Fünf Tage wollen die Meiers maximal unterwegs sein, also benötigen sie bis zu vier Hotels. Dabei wird es wohl nicht einfach mit den vier besten Hotels klappen. Daher müssen Kriterien gesucht werden, nach denen mögliche Alternativen bewertet werden.

Für die Suche nach den Alternativen kann es von Vorteil sein, einige einfache Szenarien selbst zu überlegen und daran die Suche selbst durchzuführen. Diese Handarbeit lässt sich unter anderem dadurch etwas vereinfachen, dass man die Anzahl der Übernachtungen einschränkt, bei entsprechend verkürzter Strecke.

Wortsuche (Aufgabe 3)

Für die Suche nach den verschiedenen Kriterien und den damit verbundenen Schwierigkeitsgeraden könnte es nützlich sein, wenn man eine Idee in einem Beispielrätsel umsetzt und dieses dann von einer zweiten Person lösen lässt. Diese soll dann einschätzen, ob es im Vergleich zu anderen anderen Rätseln schwerer oder einfacher war.

Für die automatische Erzeugung der Rätsel wird in einem Programm eine genaue Vorgehensweise benötigt. Bei dieser ist es hilfreich, die Überlegungen zuerst auf Papier durchzuführen. Die Möglichkeiten, wie man die Buchstaben im Programm abspeichert, sollte dabei schon berücksichtigt werden.

Würfelglück (Aufgabe 4)

Bei der Aufgabe werden immer zwei Würfel direkt miteinander verglichen, indem man ein Spiel mit ihnen simuliert. Deshalb ist es wichtig zu klären, ab wie vielen Spielen bzw. unter welchen Umständen ein Programm festlegen kann, ob einer der Würfel besser ist oder ob es keinen großen Unterschied gibt. Außerdem muss überlegt werden, wie man eine Reihenfolge unter den Würfeln festlegt, nachdem alle Würfel gegen jeden anderen angetreten sind.

Marktwaage (Aufgabe 5)

Als Schlüsselement dieser Aufgabe muss ein Algorithmus gefunden werden, mit dem man zu einem gewünschten Ziel die entsprechenden Gewichtsstücke und ihre Verteilung finden kann. Eine Umsetzung von Hand könnte dadurch unterstützt werden, dass man mit zwei Taschenrechnern die linke und die rechte Seite der Waage simuliert und jeweils die aufgesetzten Gewichtsstücke und das Zielgewicht auf den Seiten addiert. Beim Hinzufügen wird dieses Stück addiert, beim Entfernen subtrahiert. Mit einem dritten Taschenrechner lässt sich schnell der Unterschied der beiden Seiten erfassen. Wichtig bei dem „Ausprobieren“ ist, dass man ein eindeutiges Vorgehen hat.

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 Anzahl der Zeilen und Spalten
- Es folgen die Zeilen mit den Angaben zu den einzelnen Personen

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 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 11 erstellt.

```
1 public static void leseKarte(String dateiName) throws IOException {
2     try (BufferedReader br = new BufferedReader(new FileReader(dateiName))) {
3         // Lese Zeile mit Groessen
4         String zeileGroesse = br.readLine();
5         String[] groessen = zeileGroesse.split(" ");
6         int zeilen = Integer.parseInt(groessen[0]);
```

```

7     int spalten = Integer.parseInt(groessen[1]);
8     for (int zeilenpos = 0; zeilenpos < zeilen; zeilenpos++) {
9         // Lese die Zeile aus der Datei
10        String zeile = br.readLine();
11        String[] zellen = zeile.split(" ");
12        for (int spalte = 0; spalte < spalten; spalte++) {
13            // Dies sind nun die Werte der einzelnen Positionen.
14            // Diese müssen irgendwie gespeichert werden.
15            System.out.println(zellen[spalte]);
16        }
17    }
18
19 } catch (NumberFormatException | IOException e) {
20     // Fehler, wenn die Datei nicht gelesen werden kann.
21     System.out.println("Fehler beim Einlesen der Datei " + dateiName);
22     throw e;
23 }
24 }

```

Python

Dieser Python-Code wurde mit Python 3 erstellt.

```

1 fileName = "praeferenzen0.txt"
2 with open(fileName, "r") as f:
3     # Lese Zeile mit Groessen
4     zeileGroesse = f.readline()
5     groessen = zeileGroesse.split()
6     zeilen = int(groessen[0])
7     spalten = int(groessen[1])
8     for zeilenpos in range(zeilen):
9         # Lese Zeile aus der Datei
10        zeile = f.readline()
11        zellen = zeile.split()
12        for spalte in range(spalten):
13            # Dies sind nun die Werte der einzelnen Positionen.
14            # Diese müssen irgendwie gespeichert werden.
15            print(zellen[spalte])

```