43. Bundeswettbewerb Informatik
Anregungen für den Unterricht

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

September 2024

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 43. 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.

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). Dasselbe gilt auch für Algorithmen.

Natürlich können Sie diese Hinweise auch direkt an die Schülerinnen und Schüler weitergeben; sie sind entsprechend formuliert. Die Erfahrung zeigt aber, 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/43/`

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 programmieren zu können. Dafür können gut die Musterlösungen aus den letzten Wettbewerben herangezogen werden. So lernen die Schülerinnen und Schüler diese auch als mögliche Inspiration für die eigenen Lösungen kennen. Diese Lösungen sind unter dem Punkt Tipps auf den Seiten des BWINF zu erreichen:

`https://bwinf.de/bundeswettbewerb/tipps/`

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.

BWINF Alumni & Freunde e. V. · Hermannstädter Str. 1a · 53119 Bonn
WWW: `http://alumni.bwinf.de` · E-Mail: `vorstand@alumni.bwinf.de`

Ansprechpartner für diesen Text: Johannes Pieper · E-Mail: `bwinf@johpie.de`

Quadratisch, praktisch, grün (Junioraufgabe 1)

Bereits in der Aufgabenstellung steht der Hinweis, dass „so quadratisch wie möglich“ festzulegen ist. Hier könnte es hilfreich sein Rechtecke zu betrachten, die fast ein Quadrat sind (z. B. 9,5 x 10,5) und diese mit einem Quadrat (z. B. 10 x 10) und deutlichen Rechtecken (z. B. 25 x 4 oder 7 x 14) zu vergleichen.

Unabhängig von der Frage, welches die beste Aufteilung ist, muss man aber zuerst mögliche Aufteilungen finden. Dazu kann man sich eigene Zahlen nehmen (z. B. 160 x 70 und 21 Interessenten) und für diese alle möglichen Aufteilungen heraussuchen. Eine wichtige Zahl ist dabei die maximale Zahl der Aufteilungen des Feldes. Findet man bei dieser Suche ein strukturiertes Vorgehen, so hat man auch die Grundlage für das nötige Programm. Dieses sollte man noch an anderen eigenen Zahlen wiederholen, bevor man sich an das Programmieren macht.

Texthopsen (Junioraufgabe 2)

Bei dieser Aufgabe ist klar beschrieben, was gemacht werden muss. Deshalb hier die Empfehlung, einfach ein beliebiges (Informatik-)Buch aufschlagen und auf der Seite selbst zu zweit los hopsen.

Im zweiten Schritt kann man sich dann damit auseinandersetzen, wie in der gewählten Programmiersprache herausgefunden werden kann, was zum Beispiel der 11. Buchstabe in einem Text ist. Dann sollte man recherchieren, wie man im Programm bestimmt, wie weit man von dort aus springt. Auch ein wichtiger Teil für die Programmierung ist zu schauen, wie man den Inhalt einer Datei in das selbstgeschriebene Programm einliest.

Hopsitexte (Aufgabe 1)

Die Aufgabe hat eine sehr offene Aufgabenstellung. Lies deshalb die Aufgabenstellung sehr genau und überlege was genau in der Aufgabe zu erfüllen ist, bzw. was auch nicht. Darauf aufbauend ist zu überlegen, was genau das Programm können sollte.

Je nach gewählter Programmiersprache ist es aufwendiger für Zara ein entsprechendes Programm zu erstellen. Informiere dich in diesem Fall auch über Programmbibliotheken, die dir bei der Umsetzung helfen können.

Schwierigkeiten (Aufgabe 2)

Bei dieser Aufgabe geht es eigentlich nur darum, die verschiedenen Aufgaben in eine Reihenfolge zu sortieren. Die Schwierigkeit liegt darin, dass man finden muss wie man hier am besten sortiert, besonders unter der Berücksichtigung der Konflikte und ggf. auch der Punkte die man nicht kennt. Teilweise muss man entscheiden, was eine größere Gewichtung hat und vielleicht sind auch Aufgaben gleich schwer.

Die Empfehlung ist deshalb, für eine kleine Anzahl an Aufgaben zuvor die „alten“ Schwierigkeitsabstufungen aufzuschreiben. Hier sind oft auch selbst ausgedachte hilfreich. Anschließend wird für jede Aufgabe ein kleiner Zettel gemacht und diese dann mit den Schwierigkeitsabstufungen per Hand zu sortiert. Die dabei gefundene Strategie zum Sortieren sollte auch auf weitere Aufgaben mit Schwierigkeitsabstufungen angewandt werden, um die Strategie zu überprüfen.

Wandertag (Aufgabe 3)

Als Mensch ist man es gewohnt aus graphischen Darstellungen gut Werte ablesen zu können. Im Fall dieser Aufgabe kann man daher auch alle Wünsche der Bereiche, in denen die Strecke liegen darf, in einem Diagramm aufzeichnen. In diesem kann man dann mögliche Ergebnisse für die drei Streckenlängen ablesen.

Da eine graphische Erfassung mit einem Computer sich nur sehr umständlich lösen lässt, kann man diese Möglichkeit dafür nutzen, die Ergebnisse eines strukturierten Vorgehens zu überprüfen. Wahrscheinlich lohnt es sich, aus den Angaben der Personen viele kleine Bereiche zu erstellen. Aus diesen müssen dann nur die drei besten ausgewählt werden.

Krocket (Aufgabe 4)

Wenn man die Lage der Krockettore aufzeichnet lässt sich einfach graphisch bestimmen, ob es eine Linie gibt, die durch alle Tore führt. Dieses kann man nutzen, um die einfachen Beispieldaten zu überprüfen, ob es bei ihnen möglich ist mit einem Schlag alle Tore zu durchqueren. Auch kann man sich so eigene Beispiele basteln, die eine Lösung haben oder sicher keine.

Für die Umsetzung im Computer wird man mit den Koordinaten arbeiten müssen. Es gilt einen Algorithmus zu finden, um auf deren Grundlage eine Linie berechnen zu können. Diese wird einmal durch einen Startpunkt und durch ein Steigung angegeben. Für die Berechnung kann es ggf. hilfreich sein, die Steigung durch die Änderung der x- und der y-Koordinate anzugeben. Dabei wird beschrieben, um wie weit sich die Punkte auf der Linie nach links oder rechts versetzen (x-Änderung), wenn man Lage auf der Linie nach oben oder unten verändert (y-Änderung). Es könnte möglich sein, dass man damit arbeitet, dass man mögliche Bereiche angibt, in denen eine Änderung der einen Ache für eine feste Änderung der anderen Achse liegen darf.

Das ägyptische Grabmal (Aufgabe 5)

Bei dem Grabmal kann sich eine Person immer nur dann bewegen, wenn mindestens zwei benachbarte Felder frei sind, wobei die Person sich in einem der beiden befindet. Auch wenn es in der Aufgabe nicht beschrieben ist, so wird es wahrscheinlich bei einigen Eingaben der Fall sein, dass die Person auch zurückgehen muss, um dann wieder ein Teil nach vorne gehen zu können.

Eine Möglichkeit sich an die Lösung der Aufgabe heranzutasten ist es also, aufzuschreiben, wann es wo überhaupt Bewegungsmöglichkeiten für die Person gibt. Für kleinere Beispiele lässt sich dieses auch per Hand aufschreiben. Im weiteren Schritt muss dann ein systematisches Vorgehen erarbeitet werden, wie man herausfindet, welche dieser Möglichkeiten man überhaupt nutzen kann und in welcher Kombination.