



**INFORMATIK-BIBER SCHWEIZ
 CASTOR INFORMATIQUE SUISSE
 CASTORO INFORMATICO SVIZZERA**

Aufgaben und Lösungen 2022

Schuljahre 3/4

<https://www.informatik-biber.ch/>

Herausgeber:

Susanne Datzko, Nora A. Escherle,
 Jean-Philippe Pellet

010100110101011001001001
 010000010010110101010011
 010100110100100101000101
 001011010101001101010011
 010010010100100100100001

SV!A

www.svia-ssie-ssii.ch
 schweizerischerverein für informatik in
 erausbildung // société suisse pour l'infor
 matique dans l'enseignement // società sviz
 zera per l'informatica nell'insegnamento



Mitarbeit Informatik-Biber 2022

Masiar Babazadeh, Susanne Datzko, Jean-Philippe Pellet, Giovanni Serafini, Bernadette Spieler

Projektleitung: Nora A. Escherle

Herzlichen Dank für die Aufgabenentwicklung für den Schweizer-Wettbewerb an:

Juraj Hromkovič, Christian Datzko, Jens Gallenbacher, Regula Lacher: ETH Zürich, Ausbildunges- und Beratungszentrum für Informatikunterricht

Tobias Berner: Pädagogische Hochschule Zürich

Waël Almoman: Collège Voltaire

Die Aufgabenauswahl wurde erstellt in Zusammenarbeit mit den Organisatoren von Bebras in Deutschland, Österreich, Ungarn, Slowakei und Litauen. Besonders danken wir:

Valentina Dagienė, Tomas Šiaulys, Vaidotas Kinčius: Bebras.org

Wolfgang Pohl, Hannes Endreß, Ulrich Kiesmüller, Kirsten Schlüter, Michael Weigend: Bundesweite Informatikwettbewerbe (BWINF), Deutschland

Wilfried Baumann, Liam Baumann, Anoki Eischer, Thomas Galler, Benjamin Hirsch, Martin Kandlhofer, Katharina Resch-Schobel: Österreichische Computer Gesellschaft

Gerald Futschek, Florentina Voboril: Technische Universität Wien

Zsuzsa Pluhár: ELTE Informatikai Kar, Ungarn

Michal Winzcer: Comenius University, Slowakei

Die Online-Version des Wettbewerbs wurde auf cuttle.org realisiert. Für die gute Zusammenarbeit danken wir:

Eljakim Schrijvers, Justina Dauksaite, Dave Oostendorp, Alieke Stijf, Kyra Willekes, Jo-Ann Bolten: cuttle.org, Niederlande

Chris Roffey: UK Bebras Administrator, Vereinigtes Königreich

Für den Support während den Wettbewerbswochen danken wir:

Hanspeter Erni: Schulleitung Sekundarschule Rickenbach

Christoph Frei: Chragokyberneticks (Logo Informatik-Biber Schweiz)

Dr. Andrea Leu, Maggie Winter, Lena Frölich: Senarclens Leu + Partner AG

Die deutschsprachige Fassung der Aufgaben wurde ähnlich auch in Deutschland und Österreich verwendet.

Die französischsprachige Übersetzung wurde von Elsa Pellet und die italienischsprachige Übersetzung von Christian Giang erstellt.



INFORMATIK-BIBER SCHWEIZ
CASTOR INFORMATIQUE SUISSE
CASTORO INFORMATICO SVIZZERA

Der Informatik-Biber 2022 wurde vom Schweizerischen Verein für Informatik in der Ausbildung (SVIA) durchgeführt und massgeblich von der Hasler Stiftung unterstützt. Wettbewerbssponsoren sind das Amt für Wirtschaft und Arbeit des Kantons Zürich sowie die UBS.

Dieses Aufgabenheft wurde am 22. November 2023 mit dem Textsatzsystem $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ erstellt. Wir bedanken uns bei Christian Datzko für die Entwicklung und langjährige Pflege des Systems zum Generieren der 36 Versionen dieser Broschüre (nach Sprachen und Schulstufen). Das System wurde analog zum Vorgänger-System neu programmiert, welches ab 2014 gemeinsam mit Ivo Blöchliger entwickelt wurde. Jean-Philippe Pellet danken wir für die Entwicklung der **bebras** Toolchain, die seit 2020 für die automatisierte Konvertierung der Markdown- und YAML-Quelldokumente verwendet wird.

Hinweis: Alle Links wurden am 1. Dezember 2022 geprüft.



Die Aufgaben sind lizenziert unter einer Creative Commons Namensnennung – Nicht-kommerziell – Weitergabe unter gleichen Bedingungen 4.0 International Lizenz. Die Autoren sind auf S. 47 genannt.



Vorwort

Der Wettbewerb «Informatik-Biber», der in verschiedenen Ländern der Welt schon seit mehreren Jahren bestens etabliert ist, will das Interesse von Kindern und Jugendlichen an der Informatik wecken. Der Wettbewerb wird in der Schweiz in Deutsch, Französisch und Italienisch vom Schweizerischen Verein für Informatik in der Ausbildung SVIA durchgeführt und von der Hasler Stiftung unterstützt.

Der Informatik-Biber ist der Schweizer Partner der Wettbewerbs-Initiative «Bebras International Contest on Informatics and Computer Fluency» (<https://www.bebas.org/>), die in Litauen ins Leben gerufen wurde.

Der Wettbewerb wurde 2010 zum ersten Mal in der Schweiz durchgeführt. 2012 wurde zum ersten Mal der «Kleine Biber» (Stufen 3 und 4) angeboten.

Der Informatik-Biber regt Schülerinnen und Schüler an, sich aktiv mit Themen der Informatik auseinander zu setzen. Er will Berührungsängste mit dem Schulfach Informatik abbauen und das Interesse an Fragenstellungen dieses Fachs wecken. Der Wettbewerb setzt keine Anwenderkenntnisse im Umgang mit dem Computer voraus – ausser dem «Surfen» im Internet, denn der Wettbewerb findet online am Computer statt. Für die Fragen ist strukturiertes und logisches Denken, aber auch Phantasie notwendig. Die Aufgaben sind bewusst für eine weiterführende Beschäftigung mit Informatik über den Wettbewerb hinaus angelegt.

Der Informatik-Biber 2022 wurde in fünf Altersgruppen durchgeführt:

- Stufen 3 und 4 («Kleiner Biber»)
- Stufen 5 und 6
- Stufen 7 und 8
- Stufen 9 und 10
- Stufen 11 bis 13

Jede Altersgruppe erhält Aufgaben in drei Schwierigkeitsstufen: leicht, mittel und schwierig. In den Altersgruppen 3 und 4 waren 9 Aufgaben zu lösen, mit je drei Aufgaben in jeder der drei Schwierigkeitsstufen. Für die Altersklassen 5 und 6 waren es je vier Aufgaben aus jeder Schwierigkeitsstufe, also 12 insgesamt. Für die restlichen Altersklassen waren es 15 Aufgaben, also fünf Aufgaben pro Schwierigkeitsstufe.

Für jede richtige Antwort wurden Punkte gutgeschrieben, für jede falsche Antwort wurden Punkte abgezogen. Wurde die Frage nicht beantwortet, blieb das Punktekonto unverändert. Je nach Schwierigkeitsgrad wurden unterschiedlich viele Punkte gutgeschrieben beziehungsweise abgezogen:

	leicht	mittel	schwer
richtige Antwort	6 Punkte	9 Punkte	12 Punkte
falsche Antwort	−2 Punkte	−3 Punkte	−4 Punkte



Dieses international angewandte System zur Punkteverteilung soll den Anreiz zum blossen Erraten der Lösung eliminieren.

Jede Teilnehmerin und jeder Teilnehmer hatte zu Beginn 45 Punkte («Kleiner Biber»: 27 Punkte, Stufen 5 und 6: 36 Punkte) auf dem Punktekonto.

Damit waren maximal 180 Punkte («Kleiner Biber»: 108 Punkte, Stufen 5 und 6: 144 Punkte) zu erreichen, das minimale Ergebnis betrug 0 Punkte.

Bei vielen Aufgaben wurden die Antwortalternativen am Bildschirm in zufälliger Reihenfolge angezeigt. Manche Aufgaben wurden in mehreren Altersgruppen gestellt. Diese Aufgaben hatten folglich in den verschiedenen Altersgruppen unterschiedliche Schwierigkeitsstufen.

Einige Aufgaben werden für bestimmte Altersgruppen als «Bonus» angegeben: sie haben keinen Einfluss auf die Berechnung der Gesamtpunktzahl. Diese Übungen dienen vielmehr dazu, bei mehreren TeilnehmerInnen mit identischer Punktzahl zu entscheiden, wer sich für eine mögliche nächste Runde qualifiziert.

Für weitere Informationen:

SVIA-SSIE-SSII Schweizerischer Verein für Informatik in der Ausbildung

Informatik-Biber

Nora A. Escherle

<https://www.informatik-biber.ch/de/kontaktieren/>

<https://www.informatik-biber.ch/>

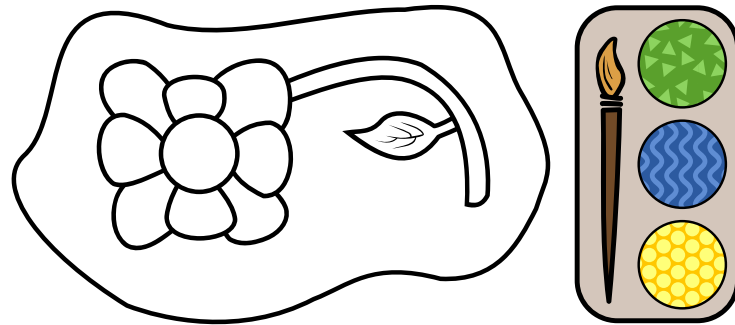


Inhaltsverzeichnis

Mitarbeit Informatik-Biber 2022	i
Vorwort	iii
Inhaltsverzeichnis	v
1. Ausmalbild	1
2. Lieblingsbonbons	5
3. Kinder lieben Bücher	7
4. Bienenwabe	11
5. Vertausche dreimal	15
6. Schildkröte und Hase	19
7. Sechsecke ausmalen	23
8. Biberburger	27
9. Matrosenkette	31
10. Hänzige Bildli	35
11. Schrauben und Muttern	39
12. Lilis Nachbarn	43
A. Aufgabenautoren	47
B. Sponsoring: Wettbewerb 2022	48
C. Weiterführende Angebote	50



1. Ausmalbild

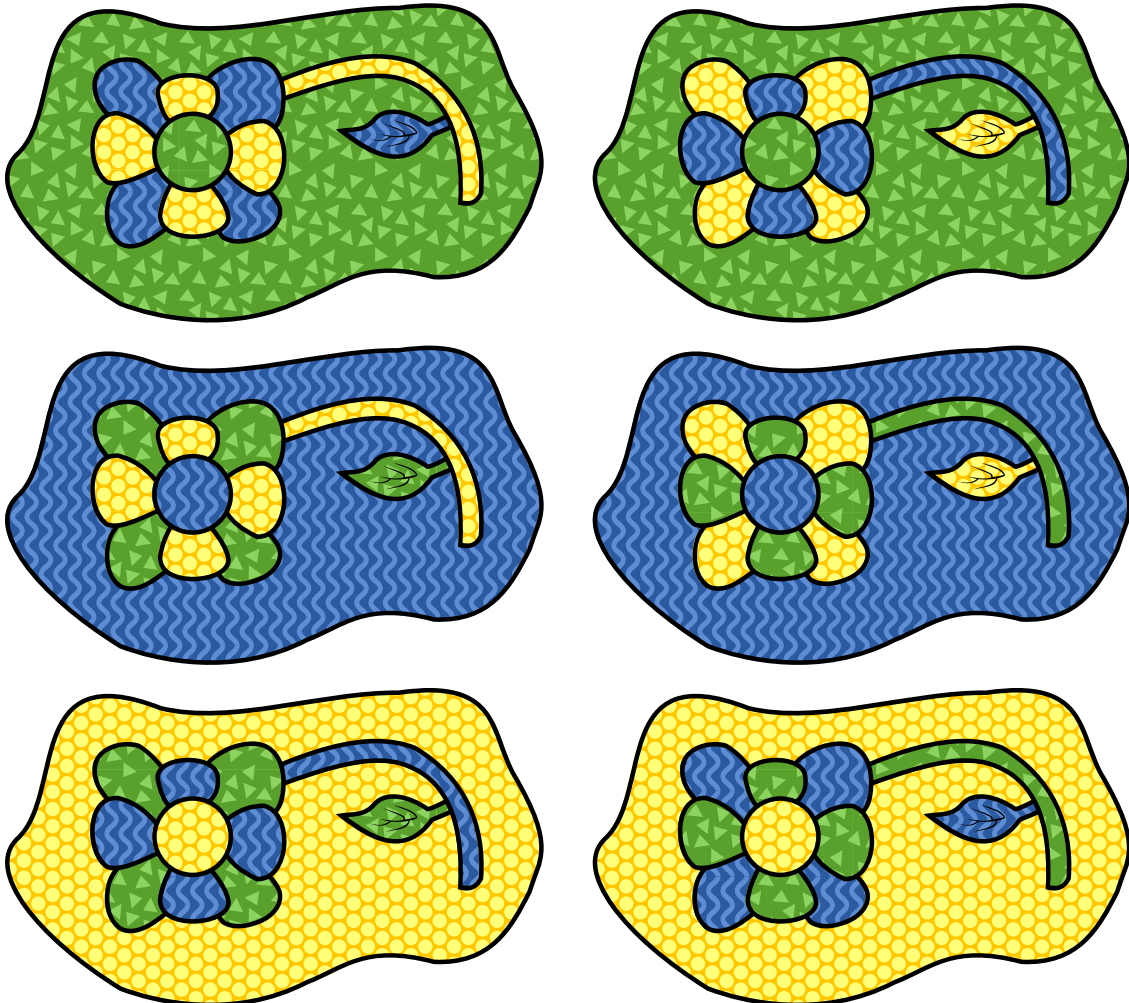


Male das Bild grün, gelb und blau aus, sodass sich keine zwei gleichfarbigen Flächen berühren.



Lösung

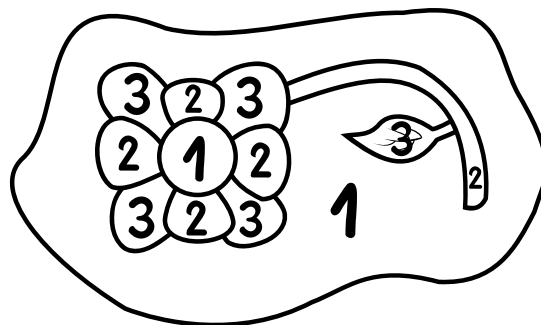
Hier siehst du alle Möglichkeiten, das Bild auszumalen.



Wie kannst du eine der Lösungen finden?

Wähle zuerst eine Farbe für die äußerste Fläche aus, z. B. Gelb. Die meisten anderen Flächen berühren die gelbe Fläche. Also müssen diese blau und grün gefärbt werden. Beginne mit einer dieser Flächen und wechsle die Farben ab. Die Mitte der Blume kann wieder gelb sein.

Dieses nummerierte Bild zeigt diese Strategie unabhängig von Farben:





Dies ist Informatik!

Diese Aufgabe ist eine Zuordnung von Farben zu Flächen, dabei müssen bestimmten Einschränkungen eingehalten werden. In der Informatik und in der Mathematik sind solche Probleme als *Färbungsprobleme* (bzw. «Graph Coloring»-Probleme) bekannt.

Färbungsprobleme haben viele Anwendungen und oft ist es wichtig, möglichst wenige Farben zu verwenden. Beispiele hierfür sind die Einteilung der Teams an einem Sportturnier, das Einteilen von Personen in Gruppen oder die Zuteilung von Frequenzen von Radiosendern. Bei Anwendungen von Färbungsproblemen gibt es meistens mehr Flächen als bei dieser Aufgabe. Es ist oft nicht mehr möglich, eine Färbung mit wenigen Farben händisch zu finden. Informatiker brauchen einen Computer, um solche Aufgaben zu lösen.

Das Färbungsproblem kann auch für Landkarten verwendet werden. Hier ist es das Ziel, eine Färbung zu finden, sodass benachbarte Länder nicht gleich gefärbt sind. Es ist für jede Landkarte möglich, so eine Färbung mit nur vier Farben zu finden. Das ist nicht einfach zu beweisen. Dies gelang erst 1976 dem Mathematiker Kenneth Appel und Wolfgang Haken. Dazu haben sie Computer verwendet, um eine Vielzahl von Ausnahmen und Gegenbeispielen zu überprüfen. Andere Menschen können das ohne Computer nicht nachvollziehen. Deshalb gibt es andere Mathematiker, die den Computereinsatz bei diesem Beweis oder allgemein beim Beweisen kritisch sehen.

Stichwörter und Webseiten

- Färbungsprobleme: [https://de.wikipedia.org/wiki/Färbung_\(Graphentheorie\)](https://de.wikipedia.org/wiki/Färbung_(Graphentheorie))
- Vorlesung über den Vier-Farben-Satz: <https://www.youtube.com/watch?v=BNx00bN6fVc>
- Vier-Farben-Satz: <https://de.wikipedia.org/wiki/Vier-Farben-Satz>





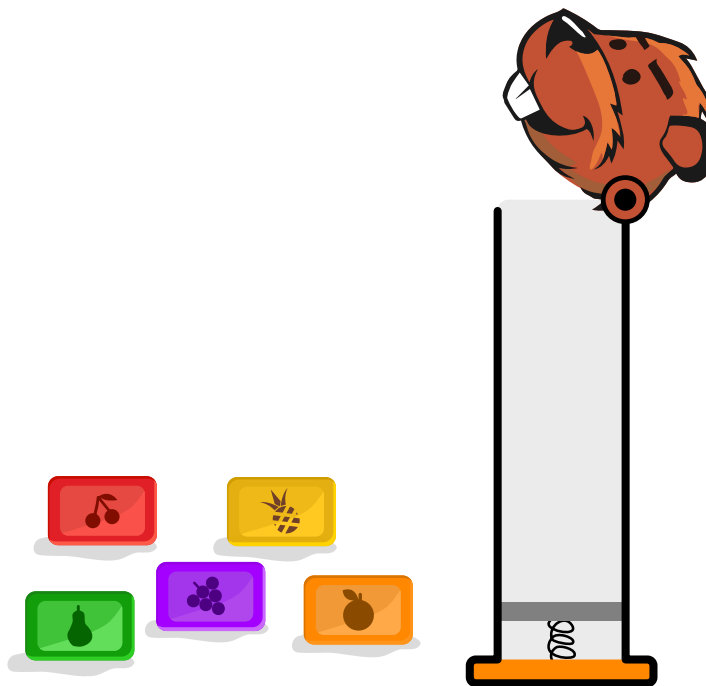
2. Lieblingsbonbons

Anna füllt fünf Bonbons in einen Spender. Danach kann sie die Bonbons so nacheinander essen, wie sie oben aus dem Spender kommen.

Sie möchte die Bonbons so nacheinander essen:



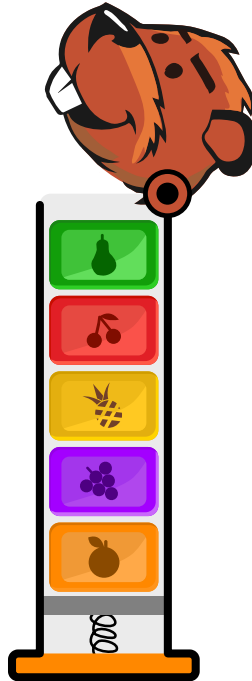
Wie muss Anna die Bonbons in den Spender füllen?





Lösung

Damit die Bonbonsorten in der richtigen Reihenfolge aus dem Spender kommen, ist es wichtig zu verstehen, dass das Bonbon, das zuerst hineingetan wird, als letztes wieder herauskommt. Das bedeutet, dass das Spender auf folgende Weise gefüllt werden muss:



Dies ist Informatik!

Wenn Anna die Bonbons einfach in der Reihenfolge ins Spender füllt, in der sie diese essen möchte, dann kommen sie genau in der umgekehrten Reihenfolge heraus. Deshalb entscheidet sich Anna zunächst für die gewünschte Reihenfolge! Anschliessend stellt sie sich vor, wie sie das Spender befüllen muss, um die Bonbons in der gewünschten Reihenfolge zu erhalten.

Auch für InformatikerInnen ist es oft wichtig, sich über die Reihenfolge Gedanken zu machen. Die Reihenfolge, die in dieser Aufgabe verwendet wird, nennt man «stack order». «Stacks», auf Deutsch *Stapelspeicher*, sind Speicherstrukturen, die Objekte in einer bestimmten Reihenfolge hinzufügen und wegnehmen: Sie arbeiten nach dem Prinzip «*Last in First out*» (LIFO). Das heisst, dass die Objekte, die als letztes hinzugefügt wurden, als erstes wieder hinausgenommen werden.

Stichwörter und Webseiten

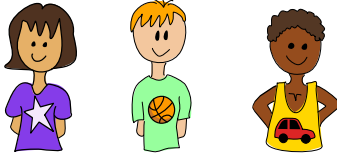
- Stapelspeicher: <https://de.wikipedia.org/wiki/Stapelspeicher>
- Last in First out (LIFO): https://de.wikipedia.org/wiki/Last_In_-_First_Out




















3. Kinder lieben Bücher

Die Kinder leihen in der Bibliothek Bücher aus. Die Bibliothek schreibt in einer Tabelle auf, wer welches Buch ausgeliehen hat.

Welches Buch haben die Kinder am häufigsten ausgeliehen?







Lösung



Die richtige Antwort ist:

Folgendes stimmt:

- Drei Kinder haben das Buch mit der Rakete ausgeliehen.
- Ein Kind hat das Buch mit der Lupe ausgeliehen.
- Zwei Kinder haben das Buch mit dem Drachen ausgeliehen.
- Ein Kind hat das Buch mit dem Hinkelstein ausgeliehen.

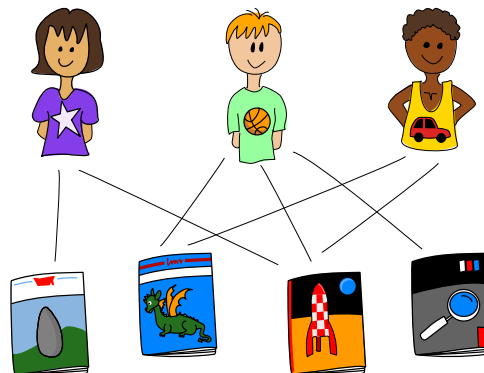


Es ist also das Buch mit der Rakete, das am häufigsten ausgeliehen wurde.

Dies ist Informatik!

Es ist schön, dass die Kinder im Informatik-Biber-Wettbewerb gerne Bücher lesen!

Aber brauchen wir wirklich eine Tabelle mit Kindern und Büchern, um die Wünsche der Kinder darzustellen? Würde es nicht auch gehen, wenn man einfach Linien zeichnen würde?





Das wäre für Menschen einfacher, aber nicht für Computer. Computer können Linien nicht gut lesen. Dafür sind sie sehr gut bei der Arbeit mit Tabellen. Wenn wir wollen, dass Computer uns bei der Arbeit helfen, wie zum Beispiel welches Kind welches Buch ausgeliehen hat, oder welcher Person welches Bankkonto gehört, dann ist es meist eine gute Idee, dies in Tabellen darzustellen.

Tabellen hat man schon vor 4000 Jahren in Babylon eingeführt, um Informationen über *Beziehungen* abzuspeichern. Diese Möglichkeit, Beziehungen abzuspeichern, macht Tabellen zu einem wichtigen Grundkonzept der *relationalen Datenbanken*.

Die Tabellen stellen die Beziehungen zwischen Dingen (oder Menschen) dar. Die Beziehungen bestimmen, wie man die Informationen in den Tabellen darstellt. Wenn es beispielsweise eine Regel gäbe, dass jedes Kind nur ein Buch ausleihen dürfte, hätte die Tabelle nur eine Zeile für jedes Kind. In unserem Beispiel mit der Bibliothek ist es in Ordnung, dass Kinder mehrere Bücher ausleihen dürfen, sie dürfen sogar auch die gleichen Bücher wie andere Kinder ausleihen. In diesem Fall braucht es diese spezielle Tabelle, die die Kinder und Bücher verbindet – und die mehrere Male das gleiche Kind und auch mehrere Male das gleiche Buch auflisten kann.

Die Ausleihtabelle ist praktisch. Wenn ein Buch fehlt, kann der Bibliothekar z. B. nachsehen, ob es verliehen wurde. Die Ausleihtabelle hat zwei Spalten und viele Zeilen. In der ersten Spalte wird das Kind eingetragen, das sich ein Buch ausleiht und in der zweiten Spalte das Buch. So kann die Frage nach dem am häufigsten verliehenen Buch ganz einfach durch Nachzählen in der zweiten Spalte beantwortet werden.

Diese Aufgabe könnte auch ein Computer erledigen. Wenn es sich um eine grosse Bibliothek mit vielen tausend Büchern handelt, dann geht es gar nicht anders! In solch einer grossen Bibliothek würde nicht nur die Ausleihtabelle geführt. Es gäbe auch eine Kundenkartei (Kundentabelle) in der alle Informationen über die Kunden wie Name, Adresse und Telefonnummer hinterlegt wären, und ein Bücherverzeichnis (Büchertabelle) mit Angaben zu den Büchern wie Autor und Titel. So bleibt die Ausleihtabelle ganz schlank weil hier nur die Beziehungen (also wer welches Buch ausgeliehen hat) zwischen den Kunden und den Büchern sind.

In der Informatik nennt man solche Tabellen relationale Datenbanken.

Stichwörter und Webseiten

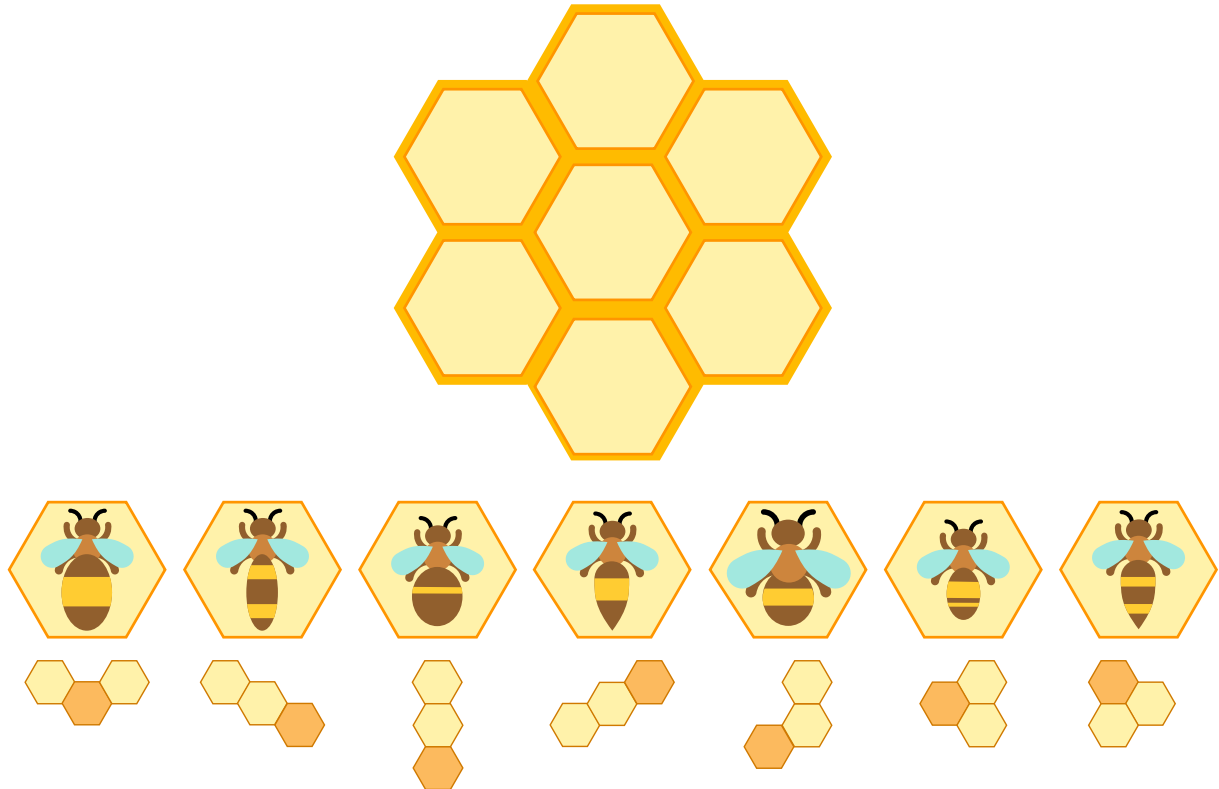
- Beziehungen (Datenbanken): https://mediencommunity.de/system/files/Beziehungen_in_Datenbanken.pdf
- relationale Datenbank: https://de.wikipedia.org/wiki/Relationale_Datenbank





4. Bienenwabe

Ein Biber braucht Hilfe, alle Bienen in seiner Bienenwabe zu versorgen.



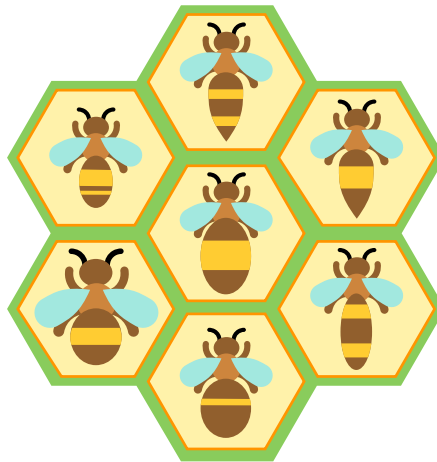
Unter jeder Biene zeigt eine Regel, in welche Zelle er die Biene versorgen darf.

Versorge die Bienen in der Bienenwabe. Beachte dabei die Regeln unterhalb der Bienen.

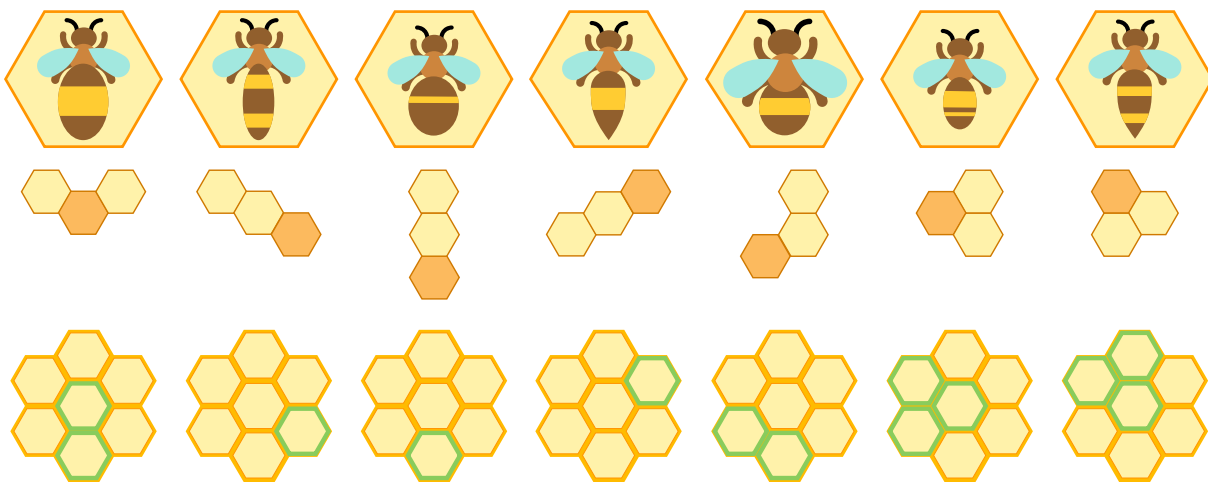


Lösung

Die Bienen können nur so in die Bienenwabe versorgt werden:



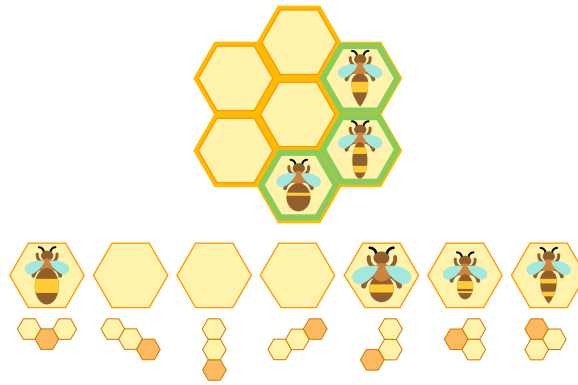
Die Aufgabe kann man durch Ausprobieren lösen. Das kann aber sehr viel Zeit in Anspruch nehmen. Um einen schnelleren Weg zu finden, schau dir die Regeln der Bienen genauer an. In der folgenden Abbildung siehst du jede Biene und die dazugehörige Regel. Jene Zellen, in welchen die Biene entsprechend ihrer Regel versorgt werden kann, sind grün umrandet.



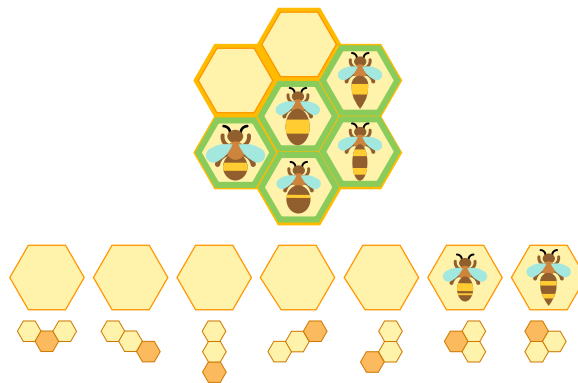
Du siehst, dass es entsprechend der Regeln für manche Bienen mehr Möglichkeiten gibt sie zu versorgen, für andere weniger. Für drei Bienen gibt es nur eine einzige Möglichkeit sie zu versorgen.

Um die Aufgabe schneller als durch Ausprobieren zu lösen, kann man wie folgt vorgehen:

Versorge zuerst die Bienen, für die nur eine Zelle der Wabe möglich ist.



Dann gibt es nur noch eine mögliche Zelle für nächsten zwei Bienen.



Auf die gleiche Weise versorgt man der Reihe nach die zwei letzten Bienen.

Dies ist Informatik!

In dieser Aufgabe müssen sieben Bienen auf sieben verschiedenen Zellen versorgt werden. Es gibt sehr viele Möglichkeiten die Bienen zu versorgen. Wenn man die Regeln berücksichtigt, reduziert sich die Anzahl der Möglichkeiten bereits um eine grosse Zahl, ist aber immer noch so hoch, dass das Ausprobieren aller Möglichkeiten einen erheblichen Aufwand bedeuten würde. Der Schlüssel zu einer schnellen Lösung der Aufgabe ist die richtige Reihenfolge. In diesem Fall starteten wir mit den am stärksten eingeschränkten Elementen, also jenen Bienen welche nur eine mögliche Zelle haben, um die Anzahl der zu untersuchenden Fälle zu begrenzen.

Einen solchen Lösungsansatz nennt man in der Informatik *Heuristik*. Durch eine geschickte Lösungsreihenfolge konnte mit wenigen Schritten die exakte Lösung gefunden werden. Bei manchen Problemen, wie zum Beispiel dem Planen einer Route zwischen verschiedenen Orten in einem Navigationssystem, geht ein heuristischer Ansatz jedoch zu Lasten der Genauigkeit. Es gibt nämlich eine riesige Anzahl an möglichen Lösungen. Um garantiert die beste Route zu finden, müsste jede mögliche Route durch das gesamte Streckennetz berechnet und verglichen werden, was mit einem riesigen Rechenaufwand verbunden wäre. Dadurch, dass man zuerst die Möglichkeiten ausprobiert, die mit hoher Wahrscheinlichkeit zu einer guten Lösung führen, kann der Rechenaufwand deutlich reduziert werden. So kann man eine gute Route in wenigen Sekunden ermitteln, anstatt die beste in Jahren zu berechnen.



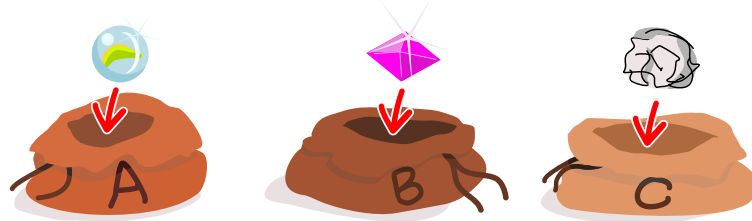
Stichwörter und Webseiten

- Heuristik: <https://de.wikipedia.org/wiki/Heuristik#Informatik>
- Routing-Heuristik: <https://de.wikipedia.org/wiki/Routenplaner#Lösungsstrategien>
- Travelling-Salesman-Problem (dt.: Problem des Handlungsreisenden):
https://de.wikipedia.org/wiki/Problem_des_Handlungsreisenden

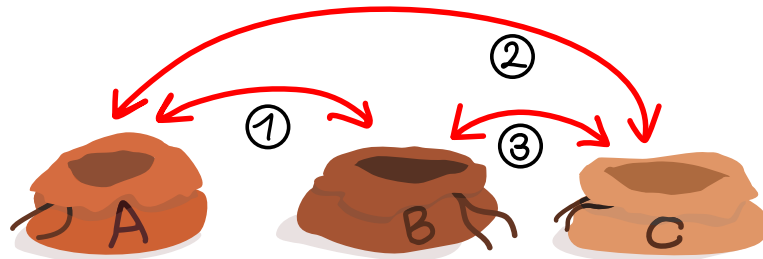


5. Vertausche dreimal

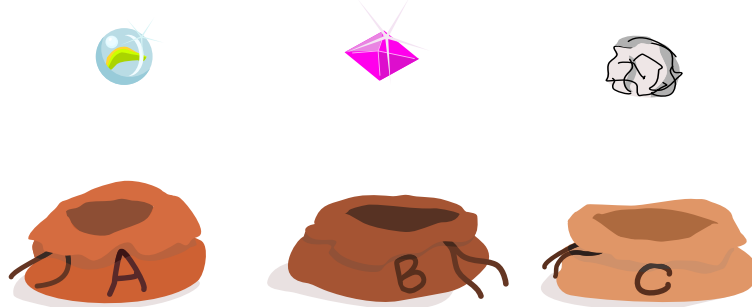
Lila legt eine Murmel in Beutel A, einen Edelstein in Beutel B und ein Stück Papier in Beutel C.



Dann vertauscht sie die Inhalte von Beutel A und Beutel B, danach die Inhalte von A und C und schliesslich vertauscht sie die Inhalte von B und C.



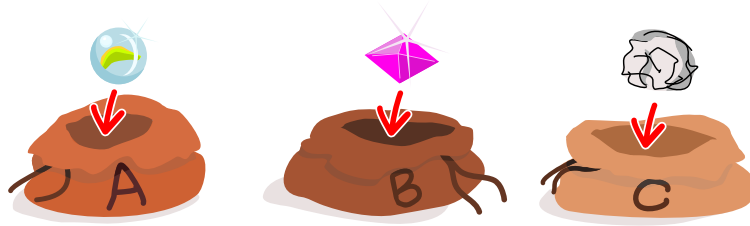
Wo befinden sich dann die 3 Dinge?





Lösung

Am Anfang haben wir diese Anordnung der 3 Dinge in den Beuteln:



Lila vertauscht die Dinge 3 Mal. Nach der ersten Vertauschung (A-B) sehen die Beutel so aus:



Nach der zweiten Vertauschung (A-C) so:



Nach der dritten und letzten Vertauschung (B-C) so:



Daher ist am Ende das Papier in A, der Edelstein in B und die Murmel in C. Dieses Ergebnis hätte man auch einfacher erreichen können, nämlich durch eine einzige Vertauschung der Inhalte von A und C.

Dies ist Informatik!

Hier geht es um Reihenfolgen von Dingen. Eine solche Reihenfolge von Dingen wird auch als Anordnung bezeichnet. Eine andere Reihenfolge stellt eine andere Anordnung dar. Eine Vertauschung ändert die Reihenfolge der Dinge und führt daher zu einer anderen Anordnung. In unserer Aufgabe haben wir am Anfang die Anordnung Murmel-Edelstein-Papier und nach den 3 Vertauschungen die Anordnung Papier-Edelstein-Murmel.

Eine interessante Frage ist, wie viele verschiedenen Anordnungen 3 Dinge haben können. Wir können es uns zunächst einmal etwas einfacher machen und nur alle Anordnungen herstellen, mit einem bestimmten Ding an erster Stelle. Für die beiden restlichen Dinge gibt es dann nur zwei Anordnungen. Wenn die Murmel an erster Stelle ist, dann sind es die beiden Anordnungen:



Murmel-Edelstein-Papier
Murmel-Papier-Edelstein

Es gibt daher auch für die beiden anderen Dinge an erster Stelle jeweils nur zwei verschiedene Anordnungen. Also gibt es noch 4 weitere Anordnungen von den 3 Dingen:

Edelstein-Murmel-Papier
Edelstein-Papier-Murmel
Papier-Murmel-Edelstein
Papier-Edelstein-Murmel

Interessant ist auch die Tatsache, dass man nur mit Vertauschungen jede beliebige Anordnung herstellen kann. Dazu sind bei n Dingen höchstens $n - 1$ Vertauschungen notwendig.

Stichwörter und Webseiten

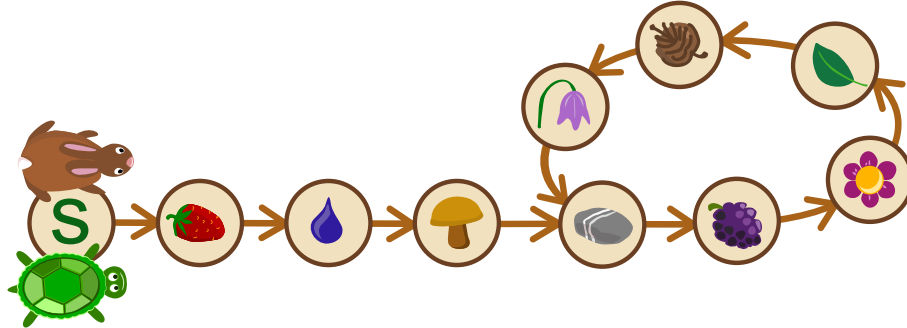
- Vertauschung:
 - https://de.wikipedia.org/wiki/Zyklische_Permutation
 - <https://www.studienkreis.de/mathematik/permutation-definition-aufgaben/>
 - <https://www.lernhelfer.de/schuelerlexikon/mathematik-abitur/artikel/permutationen>





6. Schildkröte und Hase

Eine Schildkröte 🐢 und ein Hase 🐰 machen einen Wettlauf. Sie verwenden diese Laufbahn.



Sie starten gleichzeitig auf dem Startfeld. Sie gehen von Feld zu Feld und folgen den Pfeilen.

In einer Minute geht ...

- ... die Schildkröte ein Feld vorwärts.
- ... der Hase zwei Felder vorwärts.

Auf welchem Feld treffen sich Schildkröte und Hase nach dem Start das erste Mal?

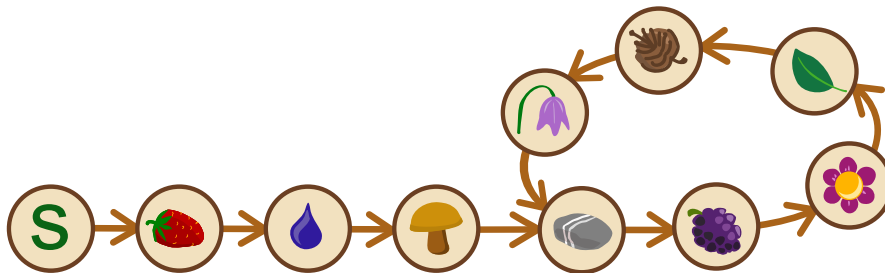


Lösung

Schildkröte und Hase treffen einander erstmals auf Feld 🌸. Man kann das leicht mit 2 Fingern nachvollziehen.

Die folgende Tabelle zeigt im Minutentakt die Felder von Schildkröte und Hase:

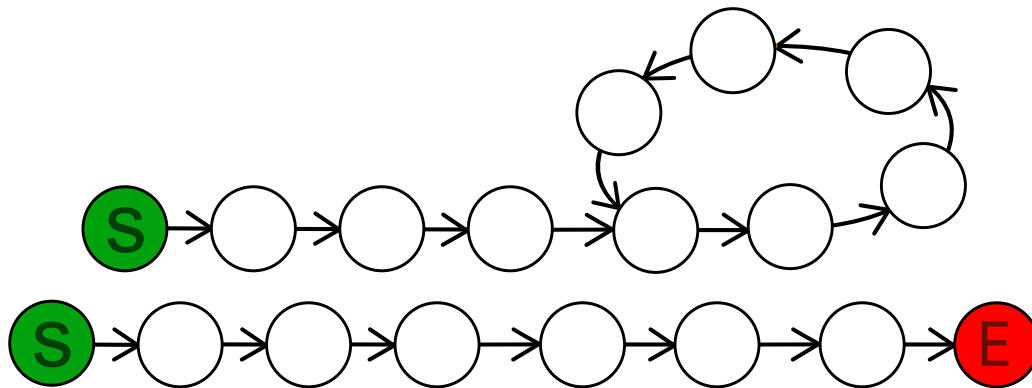
Minuten nach Start	0	1	2	3	4	5	6	7	8	9	10	11	12	13	...
	S														...
	S														...



Dies ist Informatik!

In dieser Aufgabe findet der Wettlauf auf einer besonderen Laufbahn statt. Sie besteht aus einzelnen Feldern und Pfeilen, die zum nächsten Feld zeigen. Das Besondere ist, dass die Laufbahn in einem Kreis endet, in dem die Läufer endlos laufen können. Schildkröte und Hase können sich in dieser Aufgabe nur begegnen, weil diese 6 Felder einen Kreis bzw. einen *Zyklus* bilden.

In der Informatik würde man eine Laufbahn, wie sie in der Aufgabe beschrieben ist, als *Liste* bezeichnen. Einen Kreis aus aufeinander verweisende Feldern wie in der Aufgabe würde man als *Zyklus* bezeichnen. In einer Liste verweist jeder *Knoten* auf höchstens einen anderen Knoten. Es gibt Listen mit einem Zyklus, wie in dieser Aufgabe, und Listen ohne Zyklus.



Hat eine Liste keinen Zyklus, dann besteht die Liste aus einer linearen Kette von Knoten. Dann muss es auch ein Endfeld geben, von dem kein Pfeil mehr ausgeht. Der berühmte Informatiker Robert W.



Floyd (1936–2001) hat einen Algorithmus entworfen, der einfach unterscheiden kann, ob eine Liste einen Zyklus hat oder aus einer linearen Kette besteht. Er lässt ähnlich wie in unserer Aufgabe den Hasen und die Schildkröte am Startfeld loslaufen. Wenn Schildkröte und Hase zur selben Zeit zum selben Feld kommen, gibt es einen Zyklus. In dem Moment, in dem der Hase das Endfeld oder das Feld davor erreicht, ist kein Zyklus vorhanden und der Algorithmus ist beendet.

Stichwörter und Webseiten

- Liste: [https://de.wikipedia.org/wiki/Liste_\(Datenstruktur\)](https://de.wikipedia.org/wiki/Liste_(Datenstruktur))
- Zyklus: [https://de.wikipedia.org/wiki/Zyklus_\(Graphentheorie\)](https://de.wikipedia.org/wiki/Zyklus_(Graphentheorie))
- Knoten: [https://de.wikipedia.org/wiki/Knoten_\(Graphentheorie\)](https://de.wikipedia.org/wiki/Knoten_(Graphentheorie))
- Robert W. Floyd: https://de.wikipedia.org/wiki/Robert_Floyd
- Algorithmus: <https://de.wikipedia.org/wiki/Algorithmus>



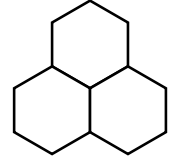


7. Sechsecke ausmalen

Sami legt weiße Sechsecke aneinander. Dann malt er sie aus, mit drei verschiedenen Farben.

Immer, wenn drei Sechsecke genau so zusammen liegen (zwei unten und eines oben in der Mitte), müssen sie am Ende ...

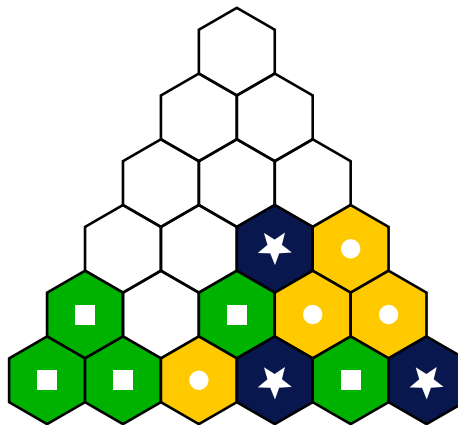
- ... alle drei die gleiche Farbe oder ...
- ... alle drei verschiedene Farben haben.



Das gefällt Sami.

Sami hat viele Sechsecke aneinander gelegt und schon einige ausgemalt.

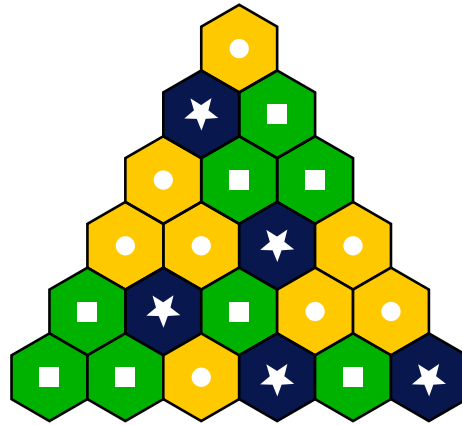
Male alle übrigen Sechsecke aus, so wie es Sami gefällt.





Lösung

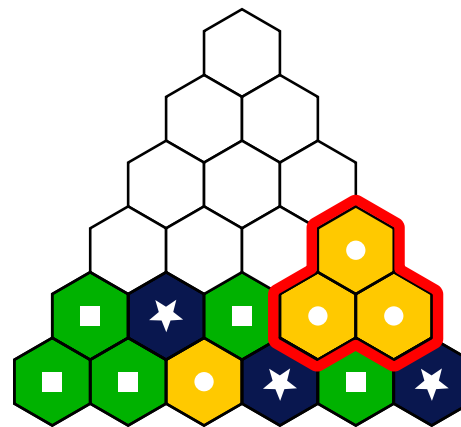
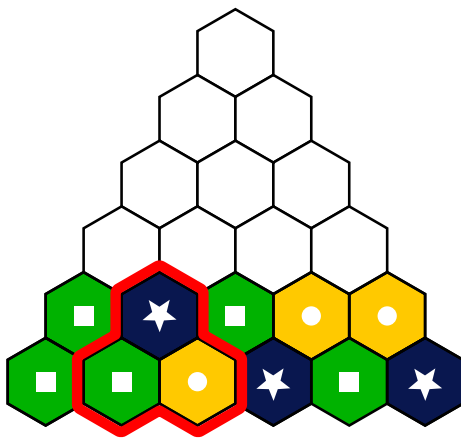
So ist es richtig:



Sobald zwei Sechsecke ausgemalt sind, die in der Sechseck-Pyramide nebeneinander liegen, steht die Farbe für das Sechseck darüber fest:

Haben beide verschiedene Farben, bekommt das Sechseck darüber die dritte Farbe. So wird zum Beispiel das unterste weisse Sechseck blau ausgemalt.

Haben beide die gleiche Farbe, bekommt das Sechseck darüber ebenfalls diese Farbe. So wird das Sechseck über den beiden gelben ebenfalls gelb ausgemalt.



So kann man die übrigen Sechsecke reihenweise, von unten nach oben, nacheinander ausmalen, so wie es Sami gefällt.

Dies ist Informatik!

Wie löst man diese Biberaufgabe? Wenn man ein Sechseck ausmalt, führt man eine Aktion aus. Um die richtige Aktion (mit der richtigen Farbe) auszuwählen, muss man sich die Sechsecke darunter anschauen und prüfen, welche *Bedingung* sie erfüllen: haben sie die gleiche Farbe oder unterschiedliche Farben. Diese Prüfung, mit anschließender Aktion, wird *wiederholt*, nämlich für jedes noch weisse Sechseck, das über zwei bereits ausgemalten Sechsecken liegt.



Aktionen, Bedingungen, Wiederholungen: Das sind die Grundbausteine eines jeden *Algorithmus*, also eines präzise beschriebenen Verfahrens, das auch als Programm für einen Computer realisiert werden kann. Beim Lösen dieser Biberaufgabe hast du also einen Algorithmus erfunden. Das ist eine der wichtigsten Aufgaben von Informatikerinnen und Informatikern: Algorithmen zu erfinden oder bereits erfundene Algorithmen nutzen und in Programme für Computer umzusetzen, um Aufgaben und Probleme mit automatischer Informationsverarbeitung zu lösen.

Stichwörter und Webseiten

- Algorithmus: <https://de.wikipedia.org/wiki/Algorithmus>
- bedingte Anweisung:
https://de.wikipedia.org/wiki/Bedingte_Anweisung_und_Verzweigung
- Schleife: [https://de.wikipedia.org/wiki/Kontrollstruktur#Schleife_\(Wiederholung,_Iterationsanweisung\)](https://de.wikipedia.org/wiki/Kontrollstruktur#Schleife_(Wiederholung,_Iterationsanweisung))



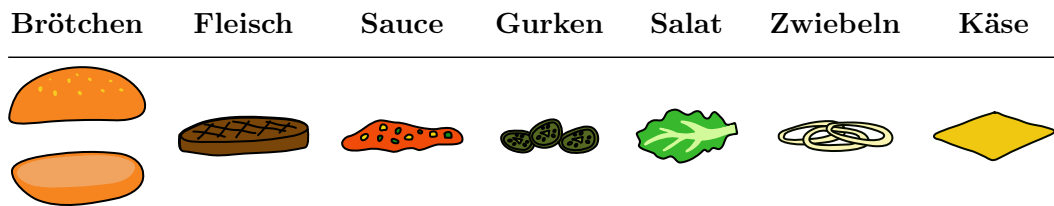


8. Biberburger

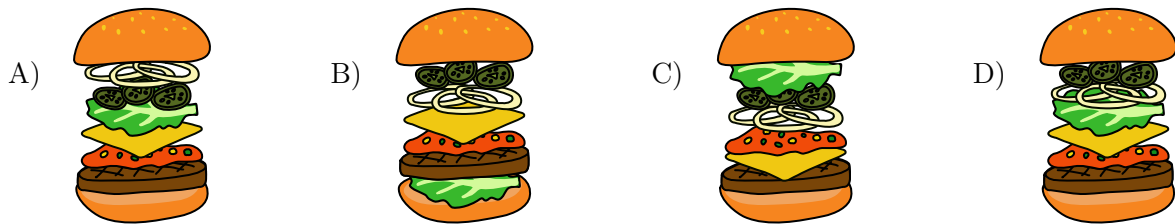
Biber Jess macht Biberburger. Er folgt dazu drei Regeln:

1. Die Sauce ist direkt auf dem Fleisch.
2. Das Fleisch und der Käse liegen unter den Gurken, dem Salat und den Zwiebeln.
3. Die Zwiebeln berühren nicht die Brötchenhälften.

Biberburger-Zutaten:



Welcher Biberburger ist nach den drei Regeln zusammengestellt?





Lösung



Die richtige Antwort ist D.

Um die Lösung zu finden, muss man bei jedem Burger prüfen, ob er so zusammengestellt ist, dass er allen drei Regeln folgt.

A) Dieser Biberburger folgt den Regeln 1 und 2. Aber die Zwiebeln berühren die obere Brötchenhälfte, also folgt er der Regel 3 nicht.

B) Dieser Biberburger folgt der Regel 1. Aber der Salat ist unter dem Fleisch und dem Käse, also wurde Regel 2 nicht befolgt.

C) Dieser Biberburger folgt Regel 2, denn das Fleisch und der Käse liegen unter den Gurken, dem Salat und den Zwiebeln. Auch folgt dieser Biberburger der Regel 3, da die Zwiebeln die Brötchenhälften nicht berühren. Allerdings ist die Sauce nicht direkt auf dem Fleisch. Somit wurde Regel 1 nicht befolgt.

D) Dieser Biberburger erfüllt alle Regeln. Biberburger D ist somit ein echter Biberburger.

Dies ist Informatik!

Die Biberburger in dieser Aufgabe sind nach drei Regeln zusammengestellt. Bei jedem der Burger, die er macht, muss Biber Jess jede der drei Regeln befolgen. Erfüllt er nur eine der Regeln nicht, ist der Burger kein Biberburger. Jede der drei Regeln ist eine Bedingung, die bei jedem Burger erfüllt sein muss, damit er ein Biberburger ist.

In der Informatik überprüft man Bedingungen (*Constraint Checking*) häufig, um herauszufinden, ob eine Lösung alle gegebenen Regeln befolgt. In dieser Überprüfung verknüpft man sämtliche Regeln (Bedingungen) mit *und*. Das bedeutet, dass alle Regeln (Bedingungen) gleichzeitig erfüllt werden müssen.

Die Prüfung, ob eine gegebene Lösung sämtliche Beschränkungen erfüllt, ist eine grundlegend andere Aufgabe als die, eine mögliche Lösung zu finden. Dies wird als *Constraint Satisfaction Problem* (dt.: Bedingungserfüllungsproblem) bezeichnet. Meistens ist es viel schwieriger, eine Lösung, die alle Bedingungen erfüllt, zu finden, als zu überprüfen, ob eine Lösung sämtliche Bedingungen erfüllt. Dies gilt sogar für einen Computer.

Stichwörter und Webseiten

- Constraintprogrammierung: <https://de.wikipedia.org/wiki/Constraintprogrammierung>



- Constraint Satisfaction Problem:
<https://de.wikipedia.org/wiki/Constraint-Satisfaction-Problem>
- und (im Zusammenhang mit der Logik):
[https://de.wikipedia.org/wiki/Konjunktion_\(Logik\)](https://de.wikipedia.org/wiki/Konjunktion_(Logik))
- NP (Komplexitätsklasse): [https://de.wikipedia.org/wiki/NP_\(Komplexitätsklasse\)](https://de.wikipedia.org/wiki/NP_(Komplexitätsklasse))

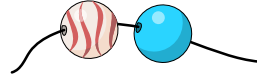




9. Matrosenkette

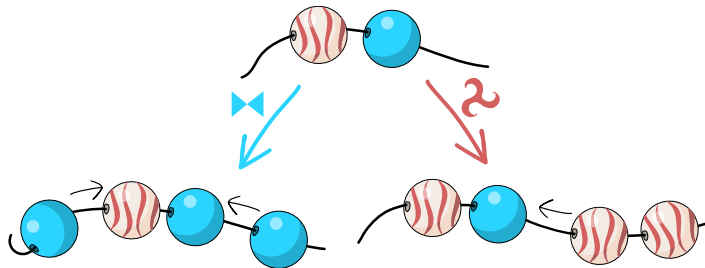
Hier ist die Anleitung für Monikas Matrosenkette mit weiss-roten Wellenperlen und einfarbigen blauen Perlen:

Du beginnst immer mit einer Wellenperle und einer blauen Perle in dieser Reihenfolge:



Dann kannst du die Matrosenkette verlängern, indem du

- an beiden Enden der Schnur jeweils eine blaue Perle hinzufügst (↔)
- oder zwei Wellenperlen am rechten Ende der Schnur hinzufügst (↷)



Diese Aktionen kannst du mehrfach durchführen, um immer längere Ketten aufzufädeln.

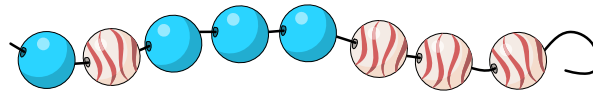
Welche der folgenden Ketten ist **keine** von Monikas Matrosenketten?

- A)
- B)
- C)
- D)














Lösung



D ist die richtige Antwort.



Die Aufgabe kannst du auf verschiedene Arten lösen.

Zum Beispiel, indem du in jeder Kette zuerst die beiden Startperlen suchst und dann eine Reihe von  - und  -Aktionen ausführst.

- Bei Kette A kannst du mit der zweiten und dritten Perle beginnen und dann die Aktionen  -  -  ausführen.
- Bei Kette B kannst du mit der dritten und vierten Perle beginnen und dann die Aktionen  -  ausführen.
- Bei Kette C kannst du mit der zweiten und dritten Perle beginnen und dann die Aktionen  -  -  ausführen.
- Wenn man jedoch die Kette D betrachtet, müssen die zweite und dritte Perle den Anfang bilden. Dann kann einmal die Aktion  ausgeführt werden, aber danach gibt es keine Aktionen mehr, um die restliche Kette zu erhalten.

Dieser Ansatz funktioniert nicht gut, wenn die Kette sehr lang ist und viele mögliche Startperlen hat. In diesem Fall kann ein dekonstruktiver Ansatz eher zum Ziel führen. Dabei entfernst du wiederholt Perlen, indem du Aktion  oder Aktion  umgekehrt ausführst, solange bis nur noch zwei Perlen übrig sind.

Eine dritte Strategie macht sich die *Parität* zunutze. Nach der Anleitung für die Matrosenkette gibt es immer eine ungerade Anzahl von einfarbigen blauen Perlen und eine ungerade Anzahl von rot-weißen Wellenperlen («ungerade Parität»). Siehst du, warum das so ist?

Kette D hat eine gerade Anzahl von beiden Arten von Perlen und kann daher keine von Monikas Matrosenketten sein.

Dies ist Informatik!

Bei dieser Aufgabe kannst du nur Perlen an den Enden der Kette auffädeln. Du kannst keine Perle in der Mitte einfügen. Du kannst auch keine Perle aus der Mitte herausnehmen, ohne zuerst die Perlen vom Ende der Kette her abzufädeln.

Diese Art von Speicherstruktur, bei der man leicht Elemente an den Enden hinzufügen und entfernen kann, aber nicht in der Mitte, wird in der Informatik *double-ended queue* oder *deque-Warteschlange* genannt (deque wird wie «Deck» ausgesprochen).

Deque-Warteschlangen können verwendet werden, um den Verlauf eines Browsers zu speichern, um Druckaufträge zu planen und auch um die Gültigkeit mathematischer Ausdrücke zu überprüfen.



Dabei kann zum Beispiel die Überprüfung auf übereinstimmende Klammern auf ganz ähnliche Weise erfolgen wie bei der Überprüfung, ob es sich bei einer Kette um eine von Monikas Matrosenkette handelt.

Stichwörter und Webseiten

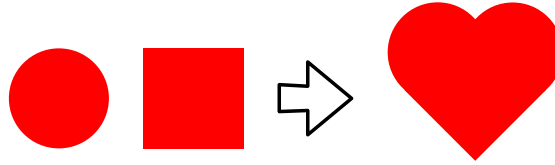
- double-ended queue: <https://de.wikipedia.org/wiki/Deque>





10. Hürzige Bildli

Tina hat zwei Formen: einen Kreis und ein Quadrat. Sie wandelt diese zu einem Herz um.



Dazu verwendet sie diese drei Umwandlungen:

- *drehe*: Eine Form beliebig weit drehen.
- *verschiebe*: Eine Form beliebig verschieben.
- *verdopple*: Eine Form verdoppeln, so dass beide an derselben Stelle bleiben.

Was hat sie in welcher Reihenfolge gemacht?

- A) *verdopple* Kreis, *drehe* Quadrat, *verschiebe* Kreis, *verschiebe* Kreis
- B) *verdopple* Quadrat, *drehe* Quadrat, *verschiebe* Quadrat, *verschiebe* Kreis
- C) *verdopple* Kreis, *drehe* Kreis, *verschiebe* Kreis, *verschiebe* Quadrat
- D) *verschiebe* Kreis, *verschiebe* Kreis, *verdopple* Kreis, *verschiebe* Quadrat






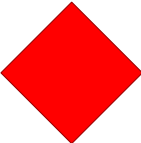

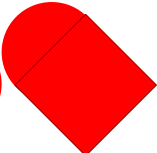
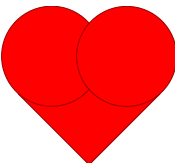


Lösung

Wenn man sich das Herz genau anschaut, stellt man fest, dass es aus zwei Kreisen und einem um $1/8$ gedrehten Quadrat besteht. Es braucht in den Umwandlungen also ein «verdopple Kreis», damit man zwei Kreise hat, und ein «drehe Quadrat», damit das Quadrat um $1/8$ gedreht werden kann. Damit fallen die Antworten B), C) und D) weg, denn:

- In der Antwort B) wird ein Quadrat verdoppelt und kein Kreis.
- In der Antwort C) wird ein Kreis gedreht, aber nicht das Quadrat.
- In der Antwort D) wird gar keine Form gedreht, also insbesondere nicht das Quadrat.

Ist aber die Antwort A) überhaupt richtig? Die Formen müssen ja noch verschoben werden! Die folgenden Umwandlungen sind vorgegeben:

- Das   ...
- ... wird durch *verdopple* Kreis zu   ...
- ... wird durch *drehe* Quadrat zu   ...
- ... wird durch *verschiebe* Kreis zu   ...
- ... wird durch *verschiebe* Kreis zu 

Deshalb ist die Antwort A) *verdopple* Kreis, *drehe* Quadrat, *verschiebe* Kreis, *verschiebe* Kreis richtig.

Dies ist Informatik!

In Bildbearbeitungsprogrammen kann man viele verschiedene Umwandlungen mit einem Bild machen. In dieser Aufgabe sind es Umwandlungen wie Drehen, Verschieben oder Verdoppeln. Das alleine genügt jedoch noch nicht: man muss dem Computer beispielsweise auch mitteilen, wie weit eine Form gedreht werden soll, oder wohin sie verschoben werden soll.

Du könntest den Weg wie man aus einem Kreis und einem Quadrat ein Herz zeichnet, natürlich auch als längeren Text beschreiben. In der Informatik ist es jedoch besser, möglichst wenige Grundumwandlungen zu verwenden, die man dann wiederholt oder unterschiedlich ausführt. Dies nennt man



Generalisieren, wenn allgemeine Lösungen aus speziellen Beispielen entwickelt werden. Solche Befehle könnten beispielsweise lauten:

- Eine Form drehen: drehe Form, wie weit
- Eine Form verschieben: verschiebe Form, wohin
- Eine Form verdoppeln: verdoppele Form

Das Bildbearbeitungsprogramm von Tina kommt einem vielleicht ungewöhnlich vor: statt dass das Bild wie bei Fotos in Form von *Pixeln* gespeichert wird, wird eine Beschreibung der Form (z. B. «Kreis, Radius 2 cm, Füllfarbe rot») gespeichert. So ist es möglich, dass zwei Formen übereinander liegen, wie die beiden Kreise, und dass nachher eines davon bewegt werden kann, ohne dass das untere überschrieben wurde. Solche Graphiken nennt man *Vektorgraphiken*. Sie werden häufig verwendet, wenn abstrakte Formen in hoher Qualität gezeichnet werden sollen. Die anderen Graphiken nennt man *Pixelgraphiken*, sie sind häufig Fotos oder fotorealistische Zeichnungen.

Stichwörter und Webseiten

- Pixel: <https://de.wikipedia.org/wiki/Pixel>
- Rastergraphik oder Pixelgraphik: <https://de.wikipedia.org/wiki/Rastergrafik>
- Vektorgraphik: <https://de.wikipedia.org/wiki/Vektorgrafik>





11. Schrauben und Muttern

Ben steht am Fließband und verarbeitet Bauteile: Muttern  und Schrauben .



Ben geht strikt nach folgendem Verfahren vor:

- Ben nimmt das nächste Bauteil vom Fließband herunter.
- Wenn Ben eine Mutter vom Fließband genommen hat, legt er sie in den Eimer.
- Wenn Ben eine Schraube vom Fließband genommen hat, nimmt er eine Mutter aus dem Eimer, schraubt sie auf die Schraube und legt das fertige Teil in den Kasten.

Bei diesem Verfahren können zwei Fehler auftreten:

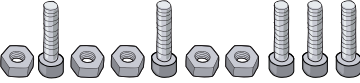
1. Ben nimmt eine Schraube vom Fließband, aber es ist keine Mutter im Eimer, die er aufschrauben könnte.
2. Ben hat alle Bauteile vom Fließband verarbeitet, aber es sind immer noch Muttern im Eimer.

Der Eimer für die Muttern ist ausreichend gross und zu Beginn leer. Welche der Folgen von Muttern und Schrauben kann Ben ohne Fehler von links nach rechts verarbeiten?

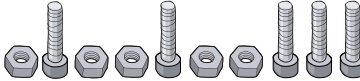

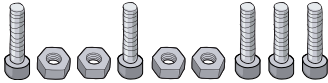




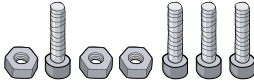


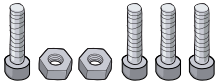


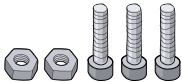


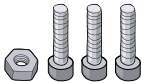



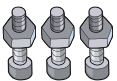


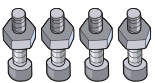


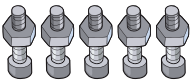
- A)
- B)
- C)
- D)





Lösung

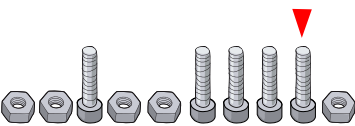

Die richtige Antwort ist C): 

Die Tabelle zeigt den Zustand des Kastens für die fertigen Teile, des Eimers für die Muttern und des Fließbands.

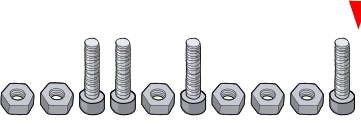
Kasten	Eimer	Fließband
<i>leer</i>	<i>leer</i>	
<i>leer</i>		
	<i>leer</i>	
		
		
		
		
		
		
		
	<i>leer</i>	<i>leer</i>

Warum sind die anderen Antworten falsch?

A)  führt zu einem Fehler an der markierten Stelle. Dann hat Ben eine Schraube aufgenommen, aber es ist keine Mutter mehr im Eimer. 

B)  führt zu einem Fehler an der markierten Stelle. Ben hat bisher 4 Muttern auf vier Schrauben geschraubt. Der Eimer ist also leer. Nun hat er aber eine fünfte Schraube aufgenommen, für die er keine Mutter mehr hat. 



D)  führt zu einem Fehler, nachdem die gesamte Folge verarbeitet worden ist. Denn es wurden 4 Muttern auf 4 Schrauben geschraubt und 2 Muttern bleiben übrig.

Dies ist Informatik!

Ben verarbeitet Bauteile, die eins nach dem anderen von dem Fließband geliefert werden. Dabei verwendet er einen grossen Eimer zum Zwischenspeichern der Muttern. Eine ähnliche Anordnung wird in der *theoretischen Informatik* als Modell für *Algorithmen* verwendet, die eine bestimmte Klasse von Problemen lösen können: *Kellerautomaten*.

Ein Kellerautomat verarbeitet Daten (Zahlen oder Zeichen), die er nach und nach als Eingabe erhält. Er besitzt einen einzigen unendlich grossen Speicher, einen Keller. Im Unterschied zum Eimer in der Aufgabe haben die Elemente im Keller eine bestimmte Reihenfolge und man kann aus einem Keller nur das Element herausnehmen, das man als letztes hineingegeben hat («last in first out», LIFO). Ein Kellerautomat kann verwendet werden, um eine *kontextfreie Sprache* zu erkennen.

In der Informatik versteht man unter einer Sprache eine Menge von Zeichenketten, die nach bestimmten Regeln geformt worden sind. Ein einfacher Typ von Sprachen sind kontextfreie Sprachen. Ein Beispiel für eine kontextfreie Sprache sind alle wohlgeformten Klammerausdrücke. Bei einem wohlgeformten Klammerausdruck wird jede geöffnete Klammer wieder geschlossen. Wohlgeformt sind z.B. ((())) und (()()). Nicht wohlgeformt sind dagegen dagegen (((() und ())((). Man kann sich die Muttern und Schrauben in der Aufgabe als öffnende und schliessende Klammern vorstellen. Dann verarbeitet Ben eine Folge von Bauteilen auf dem Fließband nur dann ohne Fehler, wenn sie einen wohlgeformten Klammerausdruck darstellt. Das Prüfen von Klammerausdrücken ist eine wichtige Aufgabe eines Compilers, der Programmtexte in ausführbare Programme übersetzt. Denn in Programmtexten der meisten Programmiersprachen kommen geschachtelte Funktionsaufrufe und arithmetische Ausdrücke mit Klammern vor.

Stichwörter und Webseiten

- Theoretische Informatik: https://de.wikipedia.org/wiki/Theoretische_Informatik
- Kellerautomat: <https://de.wikipedia.org/wiki/Kellerautomat>
- Kontextfreie Sprache: https://de.wikipedia.org/wiki/Kontextfreie_Sprache



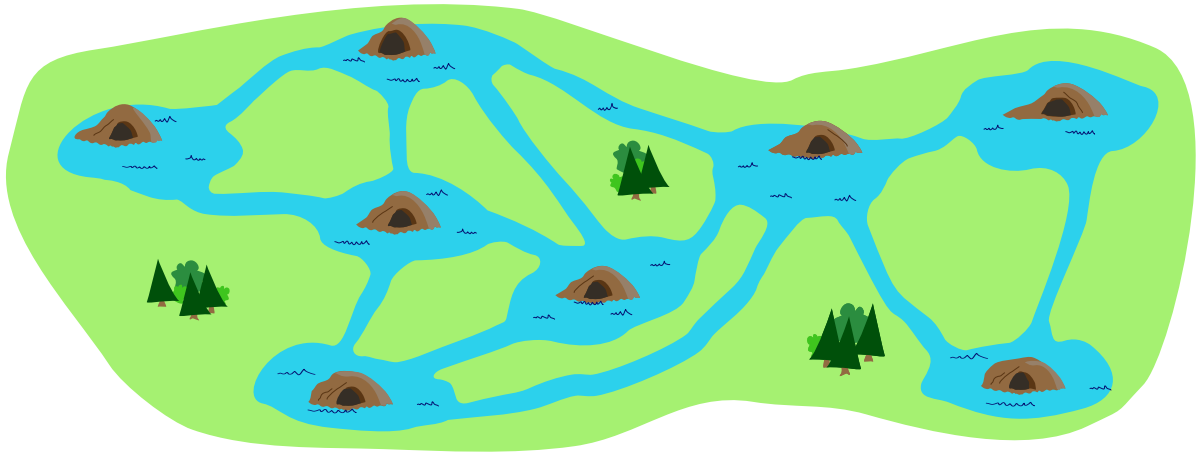


12. Lilis Nachbarn

Auf der Karte siehst du die Biberburgen von acht Bibern. Zwei Biber sind Nachbarn, wenn ein Kanal ihre Burgen direkt verbindet.

- Lili, Simon, und Peter haben je vier Nachbarn.
- Simon und Peter sind Ninas einzige Nachbarn.

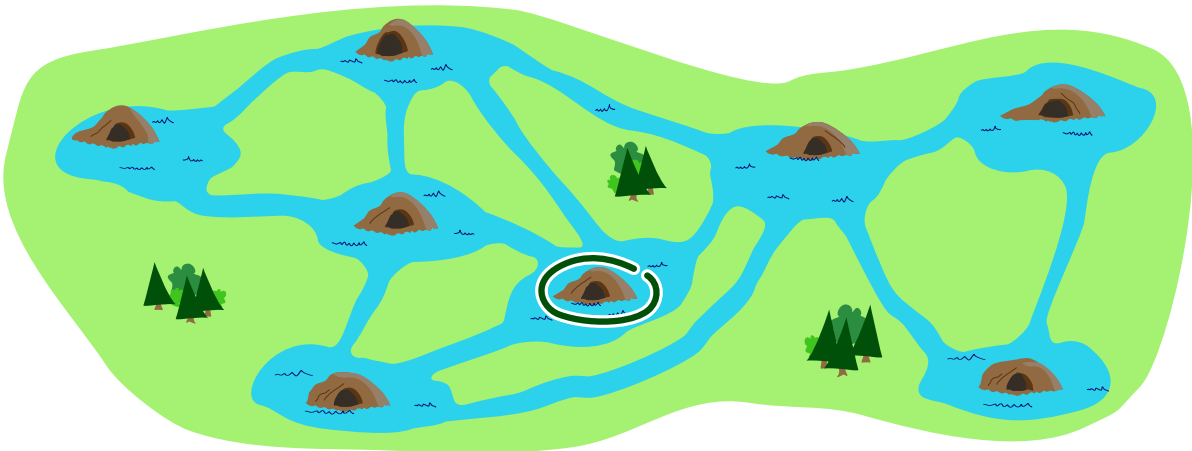
In welcher Burg wohnt Lili?



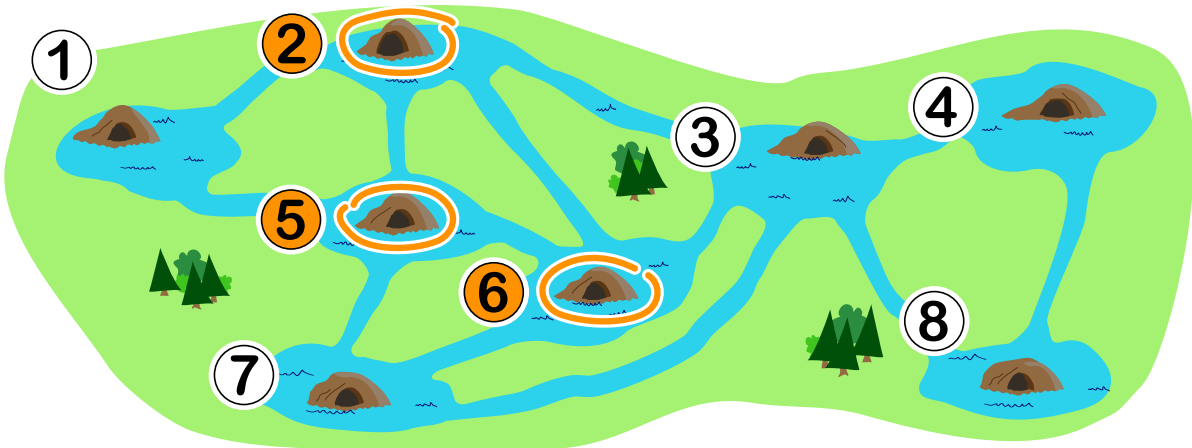


Lösung

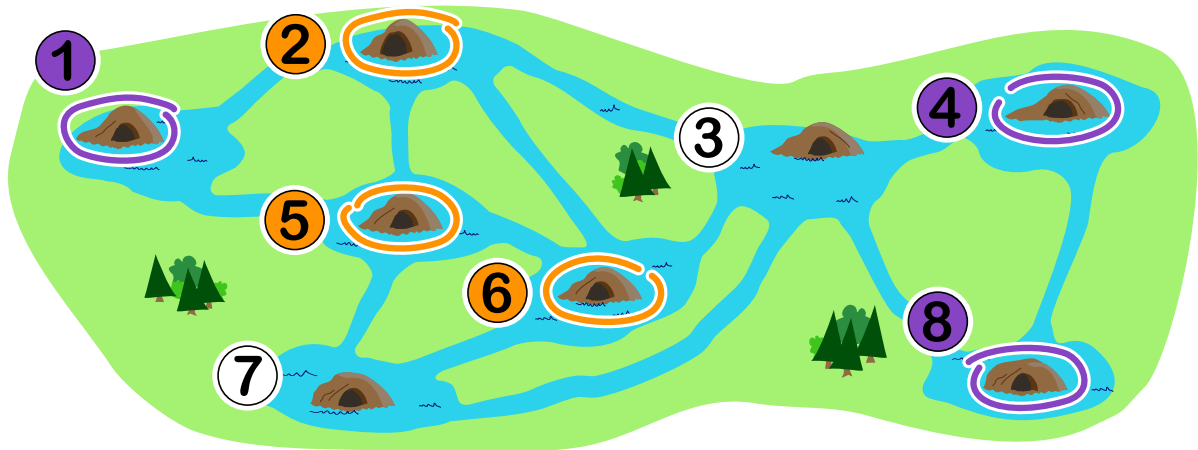
Die richtige Antwort ist:



Um das Problem zu lösen, ist es notwendig, sich auf die Kanäle zwischen den Burgen zu konzentrieren. Wir müssen die Burgen identifizieren, in denen Lili, Peter oder Simon wohnen. Da sie alle 4 Nachbarn haben, müssen von ihren Burgen je genau vier Kanäle abgehen. Es gibt drei solche Burgen: 2, 5 und 6.



Folglich leben Lili, Peter und Simon in je einer dieser drei Burgen. Nun müssen wir herausfinden, in welcher der drei Burgen Lili wohnt. Die anderen beiden Informationen beziehen sich auf Ninas Burg. Aus diesen können wir schliessen, dass von ihrer Burg genau zwei Kanäle abgehen. Also lebt Nina in einer dieser Burgen: 1, 4 oder 8.



Da wir wissen, dass Simon und Peter die beiden Nachbarn von Nina sind, können wir weiterhin folgern, dass

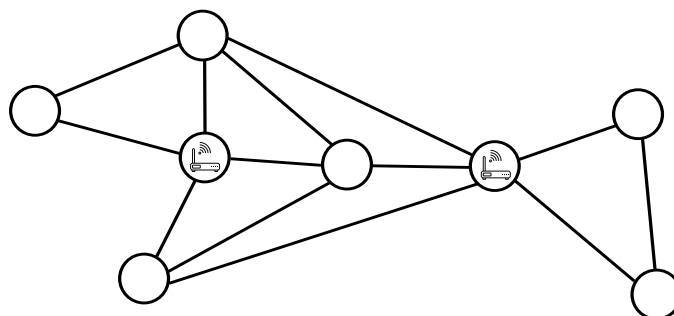
- Nina in der Burg 1 lebt
- Simon und Peter in den Burgen 5 und 7 leben (oder anders herum).

Also gibt es nur eine Burg von der vier Kanäle abgehen, die Lilis Burg sein kann. Es ist die Burg 6!

Dies ist Informatik!

In dieser Aufgabe sind jeweils zwei Biberburgen durch einen Kanal verbunden. Die Gesamtheit der Burgen und der Kanäle bildet ein Netzwerk, welches die *Beziehungen* zwischen allen Burgen aufzeigt. Ein solches Netzwerk von Beziehungen zwischen Objekten nennt man in der Informatik und der Mathematik *Graphen*. Ein Graph kann als eine *Menge* von *Knoten* betrachtet werden, die mit *Kanten* verbunden sind. In dieser Aufgabe stellen die Burgen die Knoten dar, und die Kanäle die Kanten.

Die Lehre von den Graphen nennt man *Graphentheorie*. Sie kann zur Modellierung von paarweisen Beziehungen zwischen Objekten verwendet werden. Graphen sind mathematische Modelle für netzartige Strukturen in Natur und Technik. Beispiele dafür sind soziale Strukturen, Strassennetze, Computernetze, elektrische Schaltungen, Versorgungsnetze oder chemische Moleküle. Graphen können bei der Beschreibung und Lösung von *Netzwerkproblemen* hilfreich sein, z. B. wenn es darum geht, einen guten Platz für einen Router in einem Gebäude zu finden oder sicherzustellen, dass jedes Zimmer in einem Haus ein starkes Wi-Fi-Signal hat.





Stichwörter und Webseiten

- Beziehung: [https://de.wikipedia.org/wiki/Relation_\(Datenbank\)](https://de.wikipedia.org/wiki/Relation_(Datenbank))
- Graphen: [https://de.wikipedia.org/wiki/Graph_\(Graphentheorie\)](https://de.wikipedia.org/wiki/Graph_(Graphentheorie))
- Menge: [https://de.wikipedia.org/wiki/Menge_\(Mathematik\)](https://de.wikipedia.org/wiki/Menge_(Mathematik))
- Knoten: [https://de.wikipedia.org/wiki/Knoten_\(Graphentheorie\)](https://de.wikipedia.org/wiki/Knoten_(Graphentheorie))
- Kante: [https://de.wikipedia.org/wiki/Kante_\(Graphentheorie\)](https://de.wikipedia.org/wiki/Kante_(Graphentheorie))
- Graphentheorie: <https://mathepedia.de/Graphentheorie.html>
- Netzwerkprobleme: https://www.swisseduc.ch/informatik/theoretische_informatik/hard_problems/docs/schwierigeprobleme_schueler.pdf



A. Aufgabenautoren


 Esraa Almajhad


 Leo Barichello

 Liam Baumann

 Wilfried Baumann

 Linda Björk Bergsveinsdóttir


 Tobias Berner

 Marta J. Burzanska

 Sarah Chan

 Byeonggyu Cho

 Kris Coolsaet

 Darija Dasović

 Christian Datzko


 Susanne Datzko

 Justina Dauksaite

 Nora A. Escherle

 Gerald Futschek

 Mark Edward M. Gonzales

 Yasemin Gülbahar

 Benjamin Hirsch

 Alisher Ikramov


 Thomas Ioannou

 Dauksaite Justina

 Hakin Kim

 Jihye Kim


 Seulki Kim

 Vaidotas Kinčius

 Regula Lacher

 Taina Lehtimäki

 Marielle Léonard

 Karolína Miková

 Jelena Milojkovic

 Madhavan Mukund

 Ágnes Erdősne Németh

 Zsuzsa Pluhár

 Wolfgang Pohl

 John-Paul Pretti


 Le Quang Quan

 Chris Roffey

 Kirsten Schlüter

 Bernadette Spieler

 Goran Sukovic


 Monika Tomcsányiová

 Ahto Truu

 Troy Vasiga

 Florentina Voboril

 Michael Weigend

 Kyra Willekes



B. Sponsoring: Wettbewerb 2022

HASLERSTIFTUNG

<http://www.haslerstiftung.ch/>

Stiftungszweck der Hasler Stiftung ist die Förderung der Informations- und Kommunikationstechnologie (IKT) zum Wohl und Nutzen des Denk- und Werkplatzes Schweiz. Die Stiftung will aktiv dazu beitragen, dass die Schweiz in Wissenschaft und Technologie auch in Zukunft eine führende Stellung innehat.



Kanton Zürich
Volkswirtschaftsdirektion
Amt für Wirtschaft und Arbeit

Standortförderung beim Amt für Wirtschaft und Arbeit Kanton Zürich



<http://www.ubs.com/>

Wealth Management IT and UBS Switzerland IT



<http://www.verkehrshaus.ch/>



i-factory (Verkehrshaus Luzern)

Die i-factory bietet ein anschauliches und interaktives Erproben von vier Grundtechniken der Informatik und ermöglicht damit einen Erstkontakt mit Informatik als Kulturtechnik. Im optischen Zentrum der i-factory stehen Anwendungsbeispiele zur Informatik aus dem Alltag und insbesondere aus der Verkehrswelt in Form von authentischen Bildern, Filmbeiträgen und Computer-Animationen. Diese Beispiele schlagen die Brücke zwischen der spielerischen Auseinandersetzung in der i-factory und der realen Welt.



<http://senarclens.com/>

Senarclens Leu & Partner



<http://www.abz.inf.ethz.ch/>

Ausbildungs- und Beratungszentrum für Informatikunterricht der ETH Zürich.



Scuola universitaria professionale
della Svizzera italiana

SUPSI

<http://www.hepl.ch/>

Haute école pédagogique du canton de Vaud

<http://www.supsi.ch/home/supsi.html>

La Scuola universitaria professionale della Svizzera italiana
(SUPSI)



C. Weiterführende Angebote



IT Feuer: <https://it-feuer.ch/>

In der Schweiz engagieren sich zahlreiche Organisationen für die Nachwuchsförderung in Informatik. Die Initiative «IT-Feuer» möchte diese vorhandenen Kräfte bündeln und einen Beitrag leisten, das Thema in der Öffentlichkeit schweizweit bekannter zu machen. Das IT-Feuer präsentiert eine grosse Palette an Angeboten für Lehrpersonen sowie Schüler*innen und Schulklassen.

Das Lehrmittel zum Informatik-Biber

Module

Verkehr – Optimieren

Musik – Komprimieren

Geheime Botschaften – Verschlüsseln

Internet – Routing

Apps

Auszeichnungssprachen

<http://informatik-biber.ch/einleitung/>

Das Lehrmittel zum Biber-Wettbewerb ist ein vom SVIA, dem schweizerischen Verein für Informatik in der Ausbildung, initiiertes Projekt und hat die Förderung der Informatik in der Sekundarstufe I zum Ziel.

Das Lehrmittel bringt Jugendlichen auf niederschwellige Weise Konzepte der Informatik näher und zeigt dadurch auf, dass die Informatikbranche vielseitige und spannende Berufsperspektiven bietet.

Lehrpersonen der Sekundarstufe I und weiteren interessierten Lehrkräften steht das Lehrmittel als Ressource zur Vor- und Nachbereitung des Wettbewerbs kostenlos zur Verfügung.

Die sechs Unterrichtseinheiten des Lehrmittels wurden seit Juni 2012 von der LerNetz AG in Zusammenarbeit mit dem Fachdidaktiker und Dozenten Dr. Martin Guggisberg der PH FHNW entwickelt. Das Angebot wurde zweisprachig (Deutsch und Französisch) entwickelt.



CoetryLab: <https://www.coetry-lab.org/>

Das Team des CoetryLab möchte Kindern und Jugendlichen den Zugang zum Programmieren und zu Medien ermöglichen. Das CoetryLab soll die Anlaufstelle ausserschulischen Experimentierens und Gestaltens sein und allen die Coding-Welt eröffnen. Eigene Ideen können kreativ umgesetzt und im Team oder alleine Webseiten, Apps, Games und vieles mehr entwickelt werden.



Roteco: <https://www.roteco.ch/de/>

Das ROTECO Projekt bildet eine Community für und mit Lehrpersonen, welche Schülerinnen und Schüler auf die digitale Gesellschaft vorbereiten möchten. Lehrpersonen können auf dieser Plattform Erfahrungen austauschen, erhalten Informationen zu den neusten Kursen und Workshops und finden Aktivitäten, welche sich direkt in den Unterricht integrieren lassen.



I learn it: <http://ilearnit.ch/>

In thematischen Modulen können Kinder und Jugendliche auf dieser Website einen Aspekt der Informatik auf deutsch und französisch selbständig entdecken und damit experimentieren. Derzeit sind sechs Module verfügbar.

010100110101011001001001
 010000010010110101010011
 010100110100100101000101
 001011010101001101010011
 010010010100100100100001

SV!A

www.svia-ssie-ssii.ch
 schweizerischer vereinfürinformatikind
 erausbildung//société suisse pour l'infor
 matique dans l'enseignement//società sviz
 zera per l'informaticanell'insegnamento

Werden Sie SVIA Mitglied – <http://svia-ssie-ssii.ch/svia/mitgliedschaft> und unterstützen Sie damit den Informatik-Biber.

Ordentliches Mitglied des SVIA kann werden, wer an einer schweizerischen Primarschule, Sekundarschule, Mittelschule, Berufsschule, Hochschule oder in der übrigen beruflichen Aus- und Weiterbildung unterrichtet.

Als Kollektivmitglieder können Schulen, Vereine oder andere Organisationen aufgenommen werden.