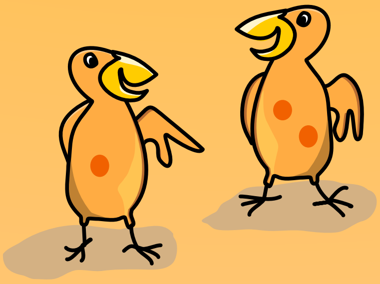
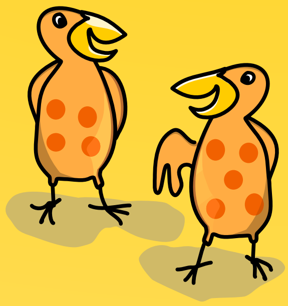
INFORMATIK-BIBER SCHWEIZ
CASTOR INFORMATIQUE SUISSE
CASTORO INFORMATICO SVIZZERA

Aufgaben und Lösungen 2021

Schuljahre 7/8



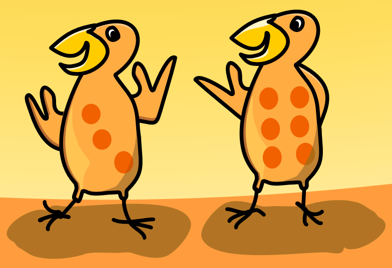
https://www.informatik-biber.ch/



Herausgeber:

Susanne Datzko, Fabian Frei,
Jean-Philippe Pellet

SVIA
www.svia-ssie-ssii.ch
schweizerischervereinfürinformatikind
erausbildung//sociétésuissepourl'infor
matiquedansl'enseignement//societàsviz
zeraperl'informaticanell'insegnamento



Mitarbeit Informatik-Biber 2021

Masiar Babazadeh, Susanne Datzko, Fabian Frei, Martin Guggisberg, Gabriel Parriaux, Jean-Philippe Pellet

Projektleitung: Nora A. Escherle

Herzlichen Dank für die Aufgabenentwicklung für den Schweizer-Wettbewerb an:
Juraj Hromkovič, Michael Barot, Christian Datzko, Jens Gallenbacher, Dennis Komm, Regula Lacher, Peter Rossmanith: ETH Zürich, Ausbildunges- und Beratungszentrum für Informatikunterricht
Bernadette Spieler: Pädagogische Hochschule Zürich

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, Arne Heijenga, Dave Oostendorp, Andrea Schrijvers, Alieke Stijf, Kyra Willekes: 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, Brigitte Manz-Brunner: Senarclens Leu + Partner AG

Diese Broschüren sind dem Andenken an Martin Guggisberg gewidmet.

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 2021 wurde vom Schweizerischen Verein für Informatik in der Ausbildung SVIA durchgeführt und von der Hasler Stiftung unterstützt.

HASLERSTIFTUNG

Dieses Aufgabenheft wurde am 24. August 2022 mit dem Textsatzsystem LaTeX 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 2021 geprüft.

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 im Rahmen des Förderprogramms FIT in IT unterstützt.

Der Informatik-Biber ist der Schweizer Partner der Wettbewerbs-Initiative «Bebras International Contest on Informatics and Computer Fluency» (`https://www.bebras.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 2021 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

In den Altersklassen 3 und 4 hatten 9 Aufgaben zu lösen, nämlich aus den drei Schwierigkeitsstufen leicht, mittel und schwer jeweils drei. 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, nämlich 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.

Für weitere Informationen:

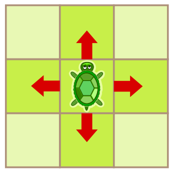
SVIA-SSIE-SSII Schweizerischer Verein für Informatik in der Ausbildung
Informatik-Biber
Nora A. Escherle

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

Inhaltsverzeichnis

1. Schildkrötenpfad

Eine Schildkröte soll verschiedene Gärten abgrasen. Jeder Garten ist in Quadrate unterteilt, die entweder mit Gras oder Steinen bedeckt sind. Die Schildkröte kann keine Steine überqueren. Sie kann sich aber von einem Grasfeld zu einem anderen Grasfeld direkt daneben bewegen.

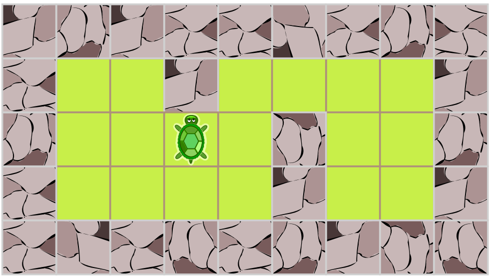


Die Schildkröte soll die Gärten vollständig abgrasen. Sie startet auf dem Feld, auf dem sie im Bild steht. Am Ende soll sie in jedem Grasfeld genau einmal gewesen sein.

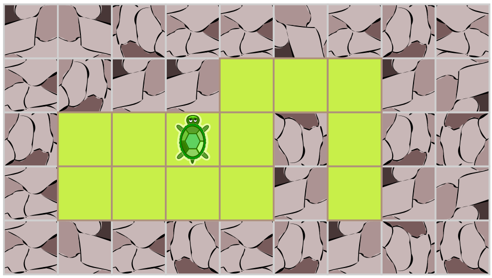
Leider kann die Schildkröte so einen der Gärten nicht vollständig abgrasen.

Welcher ist es?

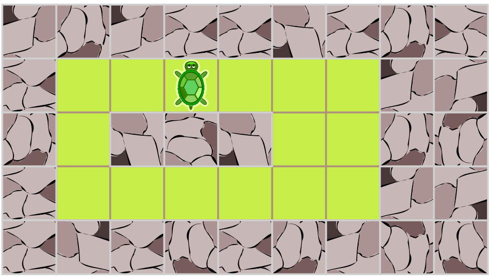
A)



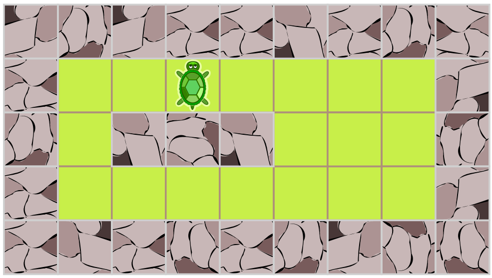
B)



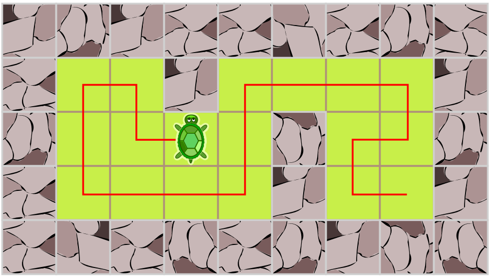
C)

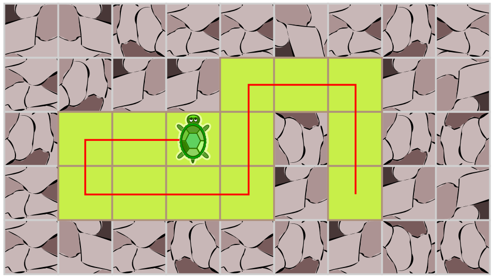


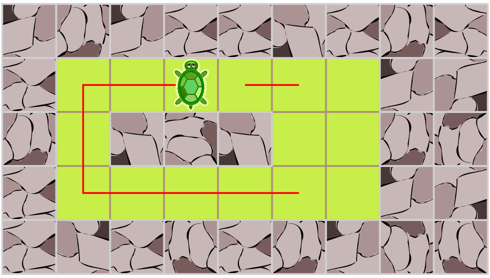
D)

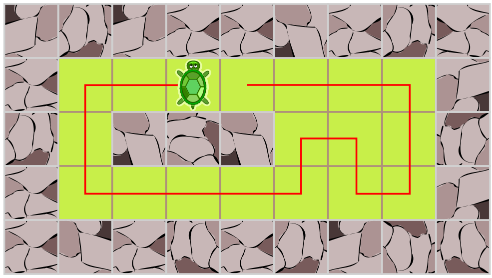


Lösung

A) 

B) 

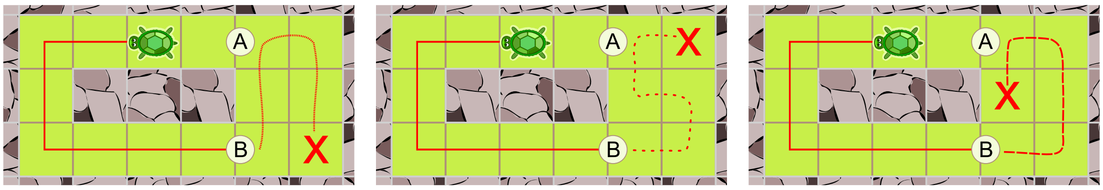
C) 

D) 

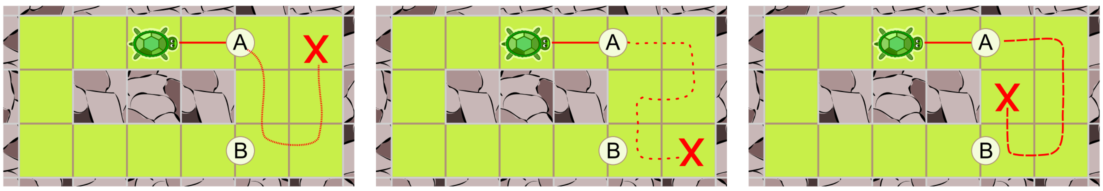
Die Gärten A, B und D kann die Schildkröte vollständig abgrasen.

Den Garten C kann die Schildkröte so nicht abgrasen. Die Schildkröte hat von ihrem Startpunkt aus nur 2 Möglichkeiten:

- Geht sie zuerst nach links, kommt sie zu Punkt B. Von dort müsste sie die 6 Felder rechts so abgrasen, dass sie am Ende Punkt A erreicht. Aber keiner der von B aus möglichen Pfade endet bei A.



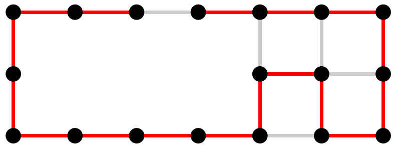
- Geht die Schildkröte zuerst nach rechts, kommt sie zu A und müsste die 6 Felder so abgrasen, dass sie am Ende Punkt B erreicht. Jetzt kann man gleich argumentieren wie vorher, man muss nur oben und unten vertauschen. Es gibt also auch so keinen geeigneten Pfad.



Dies ist Informatik!

Die Schildkröte soll einen Weg durch ihren Garten finden und dabei jedes Grasfeld genau einmal besuchen. Das Problem hinter dieser Aufgabe ist ein sogenanntes *Hamiltonpfadproblem*.

Ein Schildkrötengarten (also die Grasquadrate) kann so betrachtet werden: Jedes Grasquadrat ist ein *Knoten* (dargestellt als Punkt). Der Garten D sieht dann so aus:



Für derartige Strukturen (von Informatikern, aber auch von Mathematikern *Graphen* genannt) stellte sich im 19. Jahrhundert Sir William Rowan Hamilton die Frage, ob es einen Pfad entlang der Kanten gibt, der jeden Knoten genau einmal besucht. Einen solchen Pfad nennt man deshalb auch *Hamilton-Pfad*. Die Frage, ob es überhaupt einen Hamilton-Pfad gibt, ist im Allgemeinen sehr schwer zu lösen. Niemand kennt einen *Algorithmus*, der für beliebige Graphen effizient (in mehr oder weniger nützlicher Frist) entscheiden könnte, ob es im vorgegebenen Graphen einen Hamilton-Pfad gibt oder nicht. Wir wissen auch nicht, ob es einen solchen effizienten Algorithmus überhaupt geben kann. Dies gilt für alle sogenannten *NP-vollständigen Probleme*, wovon das Hamilton-Pfad-Problem eines der berühmtesten ist.

Stichwörter und Webseiten

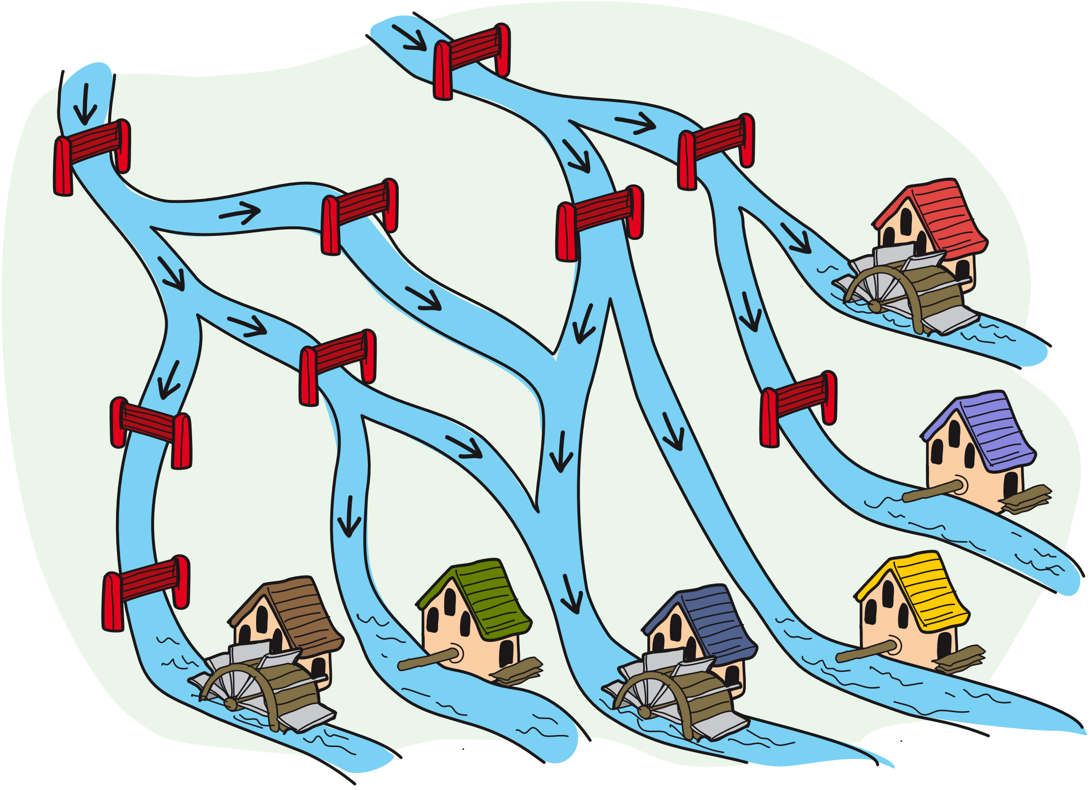
- Graph, Hamiltonpfad: `https://de.wikipedia.org/wiki/Hamiltonkreisproblem`

2. Wasser auf die Mühle

Müller Mert hat sechs Mühlen. Bei drei davon muss er noch das Mühlrad einbauen. Dafür darf kein Wasser mehr zu diesen Mühlen fliessen. Zu den anderen Mühlen soll aber weiterhin Wasser fliessen.

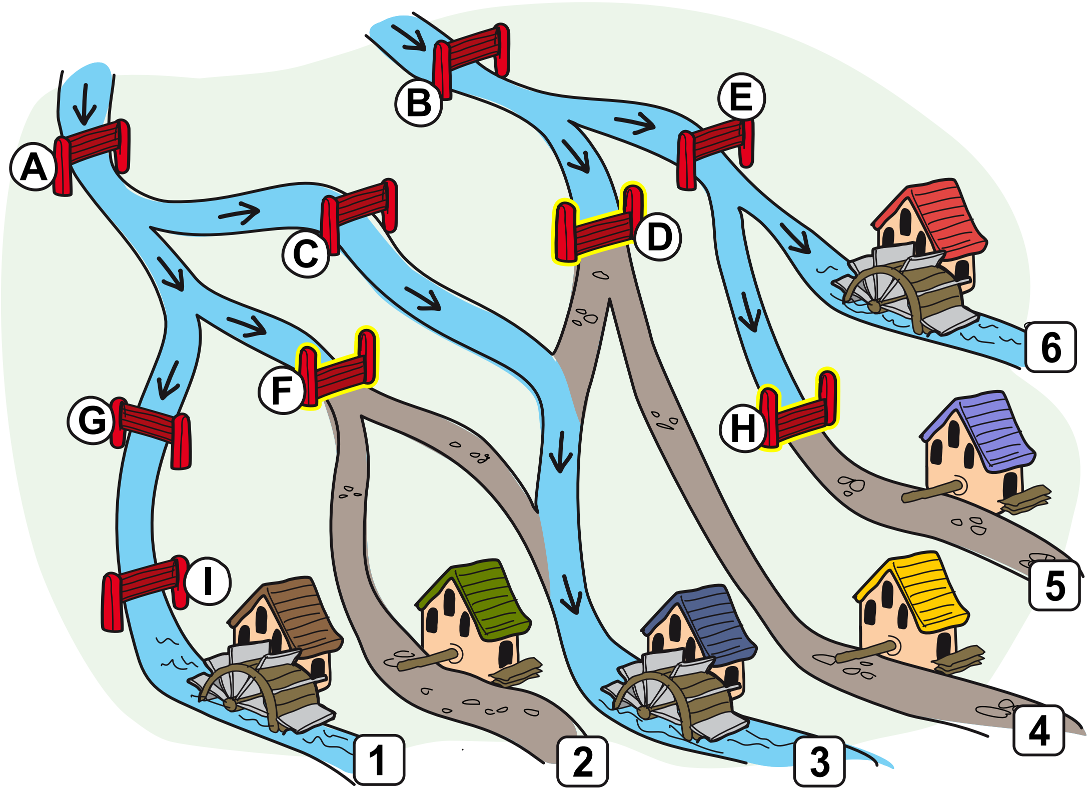
Das Wasser kann nur nach unten fliessen. Ein geschlossener Schieber stoppt das Wasser.

Welche Schieber soll Mert schliessen?



Lösung

Die richtige Antwort ist: Es sind drei Schieber zu schliessen, die in der folgenden Zeichnung mit D, F und H bezeichnet wurden.



Dies ist die einzige Möglichkeit, mit der kein Wasser mehr zu den Mühlen 2, 4 und 5 ohne Mühlrad fliesst, während die Mühlen 1, 3 und 6 weiter Wasser bekommen:

- Die Schieber A, G und I müssen alle offen bleiben, da sonst kein Wasser mehr zur Mühle 1 fliessen würde.
- Die Schieber B und E müssen ebenfalls geöffnet bleiben, da sonst kein Wasser mehr zur Mühle 6 fliessen würde.
- Weil die Schieber B und E offen bleiben, muss der Schieber H geschlossen werden, da sonst Wasser zur Mühle 5 fliessen würde.
- Weil der Schieber A offen bleibt, muss der Schieber F geschlossen werden, da sonst Wasser zur Mühle 2 fliessen würde.
- Weil der Schieber B offen bleibt, muss der Schieber D geschlossen werden, da sonst Wasser zur Mühle 4 fliessen würde.
- Weil die Schieber D und F geschlossen werden, muss der Schieber C offen bleiben, da sonst kein Wasser mehr zur Mühle 3 fliessen würde.

Dies ist Informatik!

In dieser Aufgabe wird das Fliessen des Wassers zu den Mühlen durch *Bedingungen* geregelt. Zum Beispiel fliesst genau dann Wasser zur Mühle 6, wenn die beiden Schieber B und E offen stehen. Und hier noch ein zweites, etwas komplizierteres Beispiel: Zur Mühle 3 fliesst genau dann Wasser, wenn mindestens eine oder gleich beide der folgenden Bedingungen erfüllt sind:

- Der Schieber A ist offen und einer der beiden Schieber C oder F ist offen.
- Die beiden Schieber B und D sind offen.

Solche zusammengesetzten Bedingungen werden mit den *logischen Operatoren* UND (als Symbol: $\wedge$) bzw. ODER (als Symbol: $\vee$) erzielt. Solche Operatoren verknüpfen Wahrheitswerte wie wahr oder falsch. Sind also A und B zwei Wahrheitswerte, so kann man angeben, welche Wahrheitswerte die zusammengesetzten Ausdrücke «A UND B» bzw. «A ODER B» haben:

A	**B**	**A UND B**	**A ODER B**
falsch	falsch	falsch	falsch
wahr	falsch	falsch	wahr
falsch	wahr	falsch	wahr
wahr	wahr	wahr	wahr

In der Informatik (und auch in der Mathematik) wird also die Aussage «A ODER B» auch dann als richtig angesehen, wenn beide, A und B, richtig sind. Die Aussage «Es fliesst Wasser zur Mühle 6.» ist gleichbedeutend mit:

«Der Schieber B ist offen.» UND «Der Schieber E ist offen.».

Die Aussage «Es fliesst Wasser zur Mühle 3.» aus unserem zweiten Beispiel ist gleichbedeutend mit:

(«Der Schieber A ist offen.» UND («Der Schieber C ist offen.» ODER «Der Schieber F ist offen.»))
ODER («Der Schieber B ist offen.» UND «Der Schieber D ist offen.»).

Jedes UND und jedes ODER verbindet zwei Aussagen. Die Klammer machen klar, in welcher Reihenfolge die Aussagen verbunden werden. Beim Programmieren ist das korrekte Formulieren von Bedingungen wichtig. Die Verknüpfung mit den logischen Operatoren und Klammern ist nützlich, um komplexere Bedingungen zu formulieren. Sowohl bei Verzweigung mit Hilfe von `if` als auch bei `while`-Schleifen verwenden wir Bedingungen, um den Programmablauf zu steuern.

Stichwörter und Webseiten

- Bedingte Anweisung:
  `https://de.wikipedia.org/wiki/Bedingte_Anweisung_und_Verzweigung`
- Boolsche Variable: `https://de.wikipedia.org/wiki/Boolean`
- Boolsche Operatoren: `https://de.wikipedia.org/wiki/Boolescher_Operator`

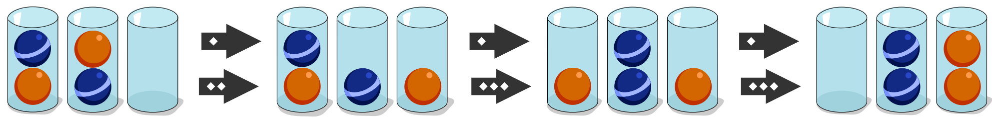
3. Kugelspiel

Die Biber möchten Kugeln nach ihrer Farbe ordnen. Am Ende sollen sich alle Kugeln in zwei Gläsern befinden. In einem Glas sollen die Kugeln dieselbe Farbe haben. Dabei sind diese drei Regeln zu befolgen:

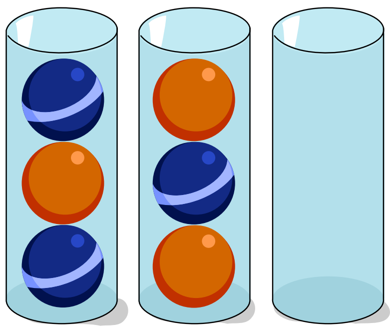
Regel 1: In einem Schritt kann nur die oberste Kugel eines Glases bewegt werden.

Regel 2: Eine Kugel kann in ein leeres Glas bewegt werden.

Regel 3: Eine Kugel kann in ein Glas bewegt werden, wenn dort noch Platz frei ist und die darunter liegende Kugel dieselbe Farbe hat.

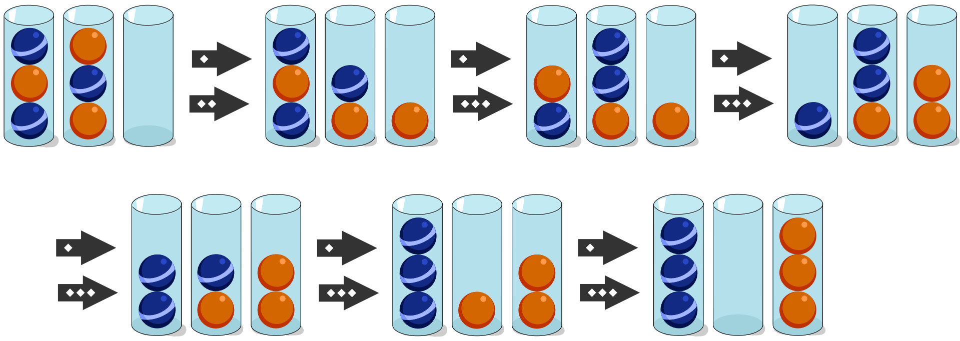


Ordne die Kugeln, indem du sie nach den drei Regeln bewegst.



Lösung

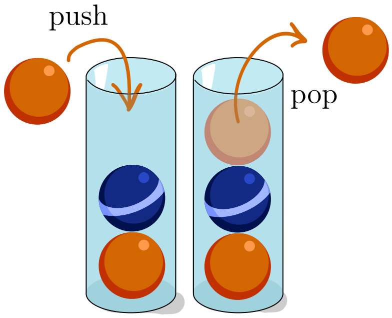
Die Kugeln können in der folgenden Reihenfolge bewegt werden:



Um die Kugeln neu zu ordnen, benötigt man mindestens 6 Schritte. Es gibt auch noch andere Möglichkeiten, die Kugeln in nur 6 Schritten neu zu ordnen.

Dies ist Informatik!

In dieser Aufgabe bewegst du die Kugeln ähnlich wie der Computer in einem *Stapelspeicher* Daten verwaltet: Er kann nur *oben ein Element* (in der Aufgabe eine Kugel) *hinzufügen* (engl. *push*) und nur das *oberste Element wieder entfernen* (engl. *pop*).



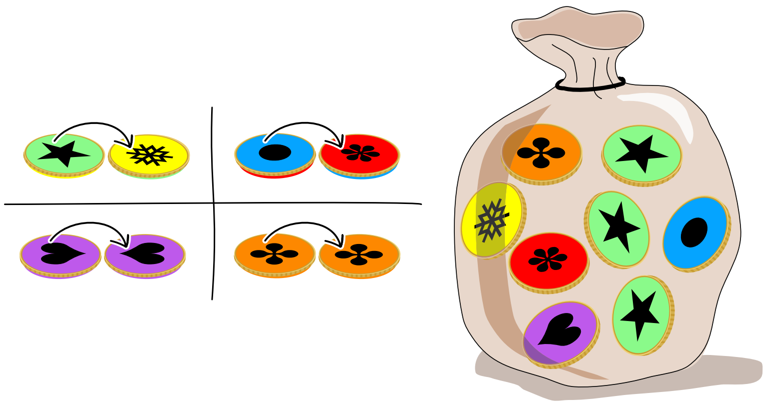
Auf die unteren Elemente kann der Computer nur zugreifen, wenn zuerst die Kugeln oberhalb entfernt werden. Und das Element, das als letztes gespeichert wurde, wird der Computer wieder als erstes entfernen. Dieses Prinzip nennen die Informatiker *Last-in-First-out-Prinzip* (kurz *LIFO*).

Stichwörter und Webseiten

- Stapelspeicher, Kellerspeicher: `https://de.wikipedia.org/wiki/Stapelspeicher`

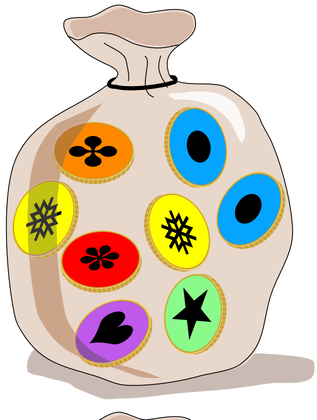
4. Sack mit Münzen

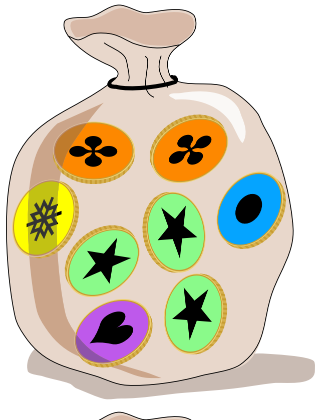
In Emils Land gibt es 4 verschiedene Arten von Münzen. Du siehst hier die beiden Seiten jeder Münzenart und auch Emils Sack mit seinen Münzen.

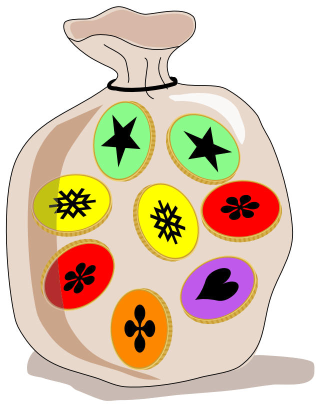


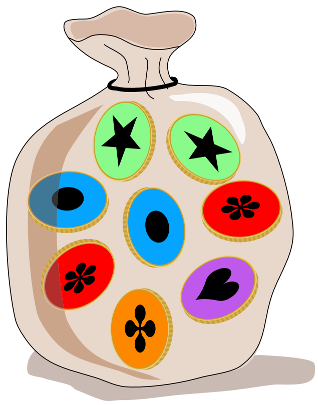
Sein Sack mit Münzen wurde nun geschüttelt.

Welcher ist Emils Sack?

A) 

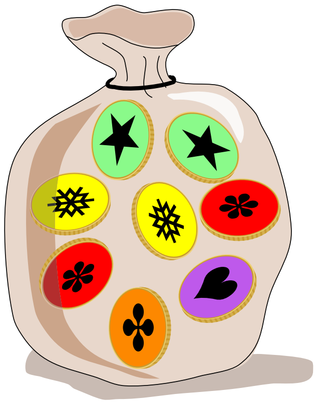
B) 

C) 

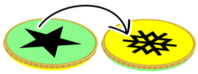
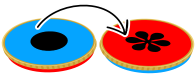
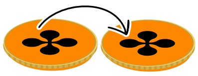
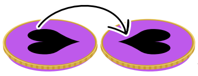
D) 

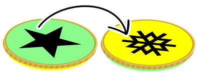
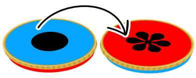
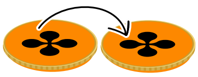
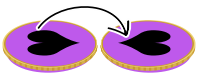
Lösung

Die richtige Antwort ist C:



In Emils Sack sind

- 4 Münzen von der Art ,
- 2 von der Art ,
- 1 von der Art  und
- 1 von der Art .

	Emils Sack	Sack A	Sack B	**Sack C**	Sack D
	**4**	3	4	**4**	2
	**2**	3	1	**2**	4
	**1**	1	2	**1**	1
	**1**	1	1	**1**	1

Nur Sack C hat für jede Münzenart die gleiche Anzahl Münzen wie Emils Sack. Daher ist C die einzige richtige Lösung.

Dies ist Informatik!

In dieser Aufgabe muss man die Münzarten erkennen, ohne dass man beide Seiten der Münzen sieht. Man hat nur eine unvollständige Information. Objekte der realen Welt werden in einem Computersystem mit ihren wesentlichen Merkmalen gespeichert. Oft genügt es, nur einen Teil dieser Merkmale zu kennen, um ein Objekt erkennen zu können. Eine Kamera in einem autonomen Fahrzeug sieht stets nur Teile der Realität und das Computersystem muss trotzdem in der Lage sein, Fahrzeuge und Verkehrsteilnehmer zu erkennen und auf die jeweilige Verkehrssituation richtig zu reagieren. Die künstliche Intelligenz in Computersystemen lernt allmählich und immer besser aus Fragmenten ganze Objekte zu erkennen, ähnlich wie Menschen.

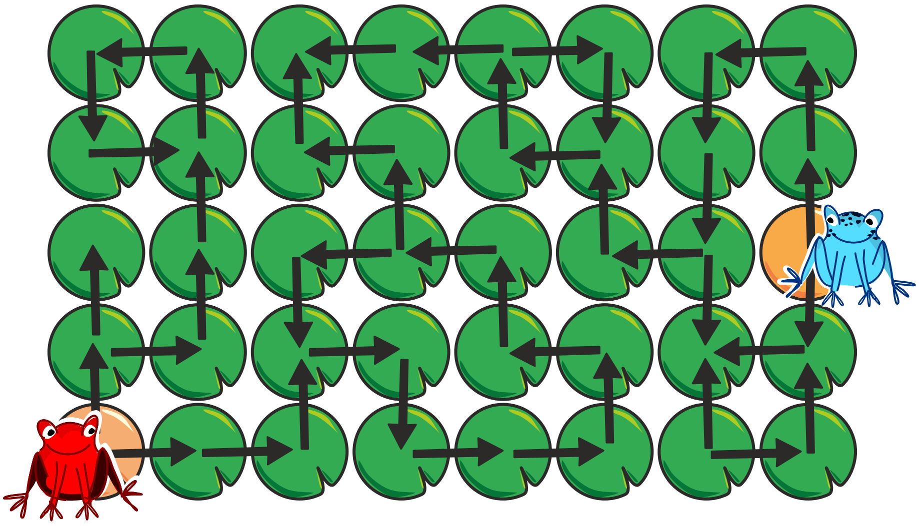
Stichwörter und Webseiten

- Multimenge: `https://de.wikipedia.org/wiki/Multimenge`
- Unstrukturierte Daten: `https://de.wikipedia.org/wiki/Unstrukturierte_Daten`

5. Treffen sie sich?

Auf einem See können zwei Frösche von Seerosenblatt zu Seerosenblatt springen – aber nur entlang der Pfeile.

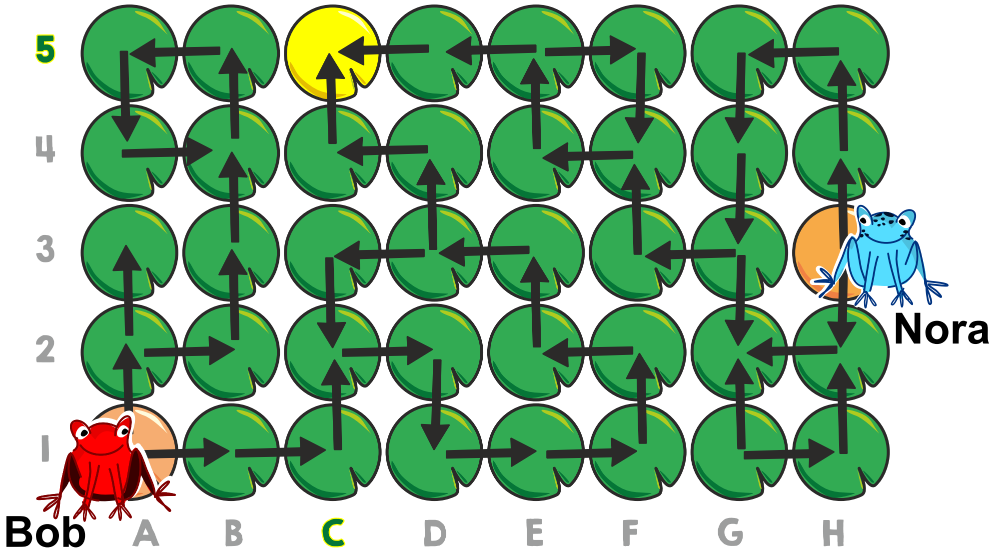
Auf welchem Seerosenblatt können sie sich treffen?



Man kann auf die Blätter klicken. Klickt man auf ein Blatt, wird dieses ausgewählt und gleichzeitig ein bereits ausgewähltes Blatt wieder deaktiviert.

Lösung

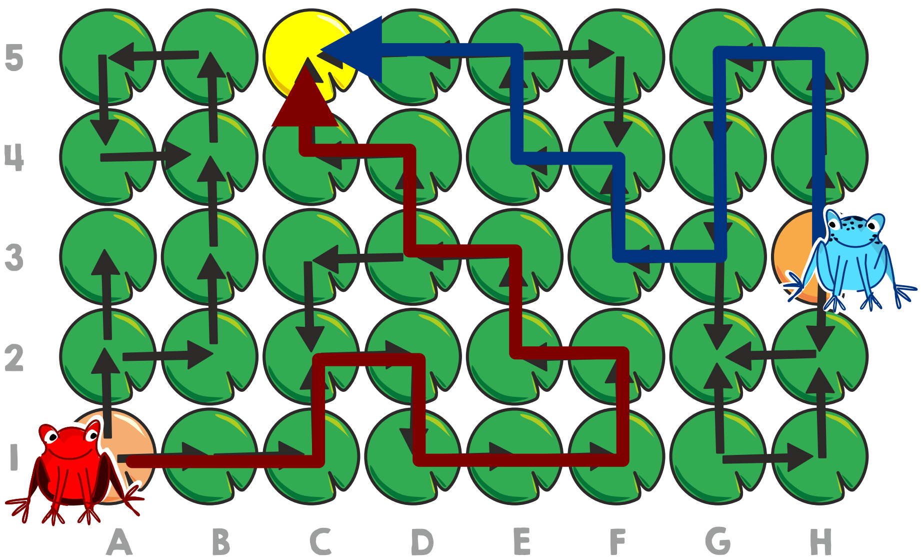
Die Frösche können sich nur auf dem Blatt C5 treffen.



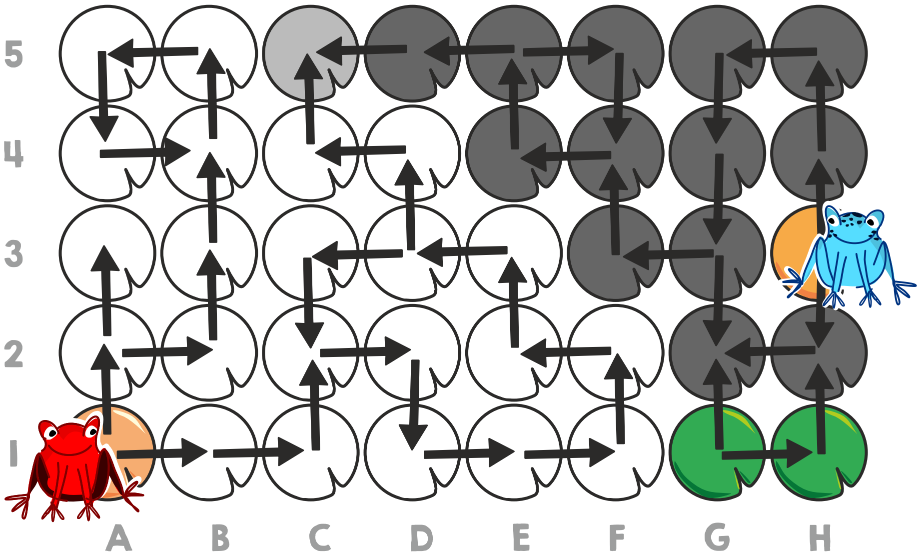
Auf seiner Startposition hat der rote Frosch Bob zwei Möglichkeiten: Geht er nach «oben», gerät er entweder in die Sackgasse A3 oder bleibt im Kreis stecken, der bei B4 beginnt. Geht er anfangs «rechts» (nach B1), kann er zunächst bis D3 weiterspringen. Dort kann er «links» in einen Kreis springen, der ihn wieder zu D3 bringt, oder nach «oben», was ihn weiter zu C5 bringt – eine weitere Sackgasse.

Der blaue Frosch Nora hat am Start auch zwei Wahlmöglichkeiten. Geht sie nach «unten», gerät sie in die Sackgasse G2. Falls sie nach «oben» startet, erreicht sie zunächst G3. Von dort kann sie entweder wieder in die Sackgasse G2 geraten oder nach «links» gehen und schliesslich E5 erreichen. Von dort geht es wieder in einen Kreis, der sie zu E5 bringt, oder zur Sackgasse in C5.

Wir wissen bereits, dass auch Bob C5 erreichen kann, also können sie sich dort treffen. Die Zeichnung zeigt die Wege, wie sie beide zu C5 gelangen können.



Das garantiert allerdings noch nicht, dass sie sich nicht auch woanders treffen könnten. Die nächste Zeichnung zeigt alle Blätter, die Bob (weiss) and Nora (dunkelgrau) erreichen können, wenn sie den Pfeilen auf jede erdenkliche Art folgen. Wir sehen, dass nur C5 von beiden erreicht werden kann.



Dies ist Informatik!

Wie kann das letzte Bild erstellt werden? Die Blätter, die von einem Frosch erreicht werden können, können mit einer *Breiten- oder Tiefensuche* gefunden werden. Dies sind zwei der wichtigsten Standardverfahren in der Informatik. Mit ihrer Hilfe kann man die dunkelgrauen und die weissen Blätter bestimmen. Schliesslich müssen nur noch die Blätter gefunden werden, welche von beiden Fröschen erreicht werden können.

Stichwörter und Webseiten

- Breitensuche: `https://de.wikipedia.org/wiki/Breitensuche`
- Tiefensuche: `https://de.wikipedia.org/wiki/Tiefensuche`

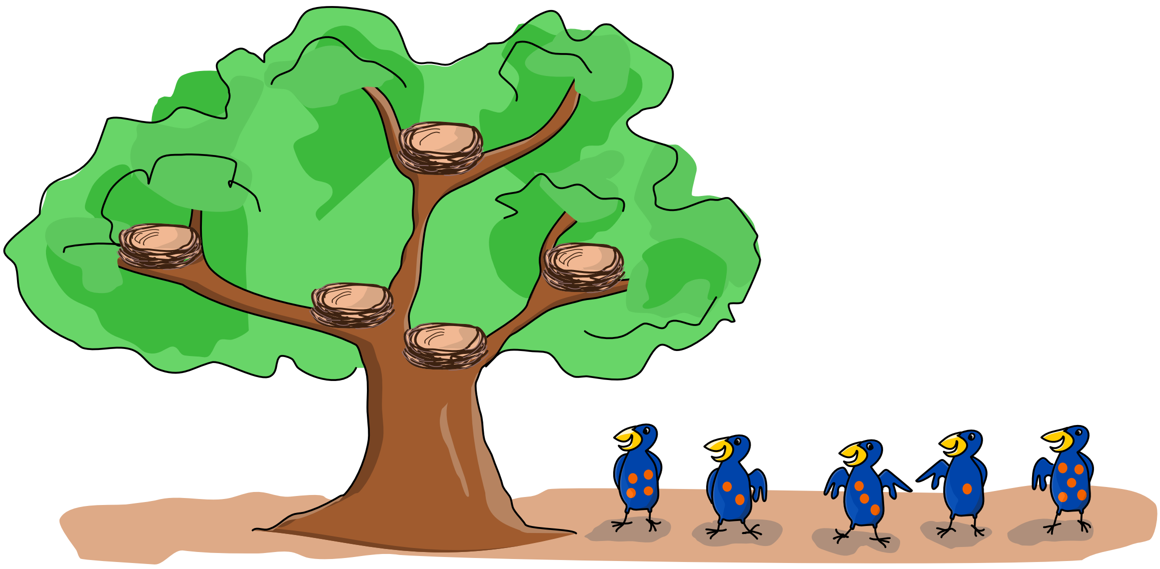
6. Dottis

Dottis sind Vögel mit Punkten. Neben einem Baum stehen fünf Dottis. Einer nach dem anderen - in der Reihenfolge von links nach rechts - klettern sie in den Baum und ziehen in die leeren Nester. Der mit den vier Punkten ist der erste. Jeder Dotti geht so vor:

Er beginnt unten am Baum. Er führt solange die folgenden Schritte aus, bis er ein leeres Nest gefunden hat:

1. Er klettert hoch, bis er ein Nest findet.
2. Wenn das Nest leer ist, dann zieht er in dieses Nest und bleibt dort.
3. Sonst klettert er weiter, und zwar, wenn der im Nest sitzende Dotti …
   - … mehr Punkte als er selbst hat, dann nach links.
   - … gleich viele oder weniger Punkte hat, dann nach rechts.

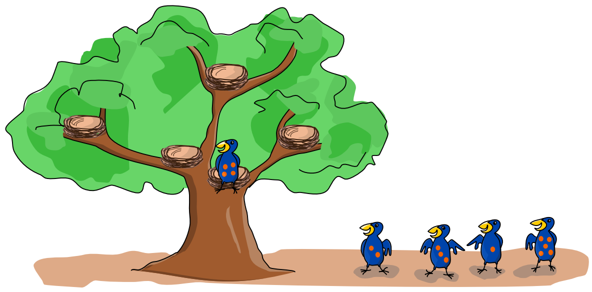
Wo sind die Dottis ganz am Ende? Setze jeden Dotti in das richtige Nest.



Lösung

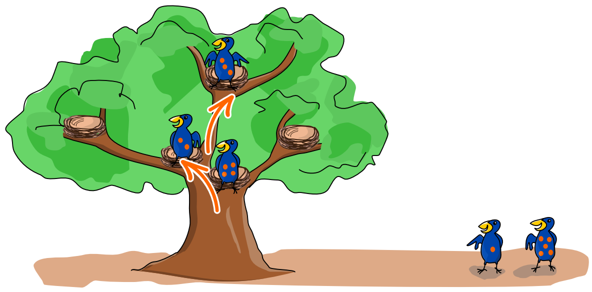
So kommt man auf die richtige Lösung:

Der erste Dotti, der mit 4 Punkten, zieht in das unterste Nest und bleibt dort.

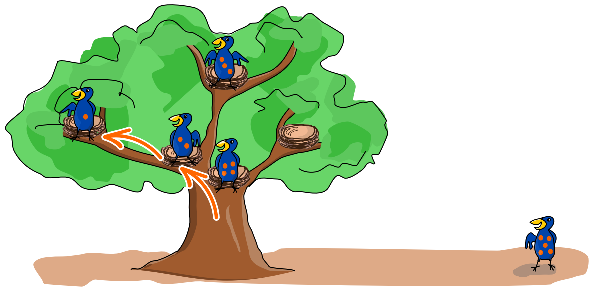


Der zweite Dotti hat 2 Punkte. Im untersten Nest sitzt der erste Dotti mit 4 Punkten. Weil 4 mehr als 2 ist, klettert der zweite Dotti nach links weiter und zieht in das erste freie Nest.

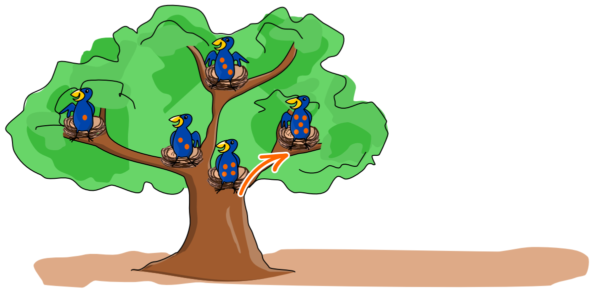
Der dritte Dotti hat 3 Punkte. Dieser klettert beim untersten Nest, wo der Dotti mit 4 Punkten sitzt, nach links, weil 4 mehr als 3 ist. Im nächsten Nest sitzt der Dotti mit 2 Punkten. Weil 3 mehr als 2 ist, klettert der dritte Dotti nach rechts weiter. Er zieht dann in das nächste freie Nest. Das ist das höchste Nest.



Der vierte Dotti hat 1 Punkt. Weil alle anderen Dottis mehr Punkte haben, klettert er bei jedem belegten Nest nach links. Er kommt dann beim Nest ganz links an und zieht dort ein.



Der letzte Dotti hat 5 Punkte. Weil kein Dotti mehr Punkte hat, klettert er bei jedem belegten Nest nach rechts. Das tut er einmal beim untersten Nest und zieht somit in das leere Nest ganz rechts.



Dies ist Informatik!

Wenn sich die Dottis nach diesem Verfahren in die Nester setzen, hat das einen interessanten Vorteil: Ein bestimmter Dotti kann dann schnell gefunden werden. Wenn der Dotti, den du suchst, weniger Punkte hat als der, auf den du gerade schaust, musst du im linken Teil des Baums weitersuchen. Ansonsten suchst du rechts weiter. Mit jeder Prüfung eines Vogels kannst du also den Suchbereich auf eine von zwei Hälften einschränken. Deshalb wirst du deinen Dotti schnell finden.

Es gibt viele Arten, auf die man Daten organisieren kann; man spricht von verschiedenen *Datenstrukturen*. Die Datenstruktur in dieser Biberaufgabe ist ein *binärer Suchbaum*. Das Wort *«binär»* kommt vom lateinischen Wort «bis» für «zweimal». Denn am Ende eines Astes (dort, wo in der Aufgabe ein Nest sitzt) führen höchstens zwei kleinere Äste weiter. Binäre Suchbäume werden in Computerprogrammen verwendet, wenn viele Daten schnell gefunden werden müssen. Sie sind meistens viel grösser als der kleine Baum in der Aufgabe. Ausserdem gibt es noch einen Unterschied: Der Baum in der Aufgabe hatte eine feste Anzahl von fünf Dottis. Dagegen kann man bei einem binären Suchbaum üblicherweise immer weiter Daten einfügen. Zum Einfügen wird einfach an das Ende eines Astes ein neuer Ast angehängt und so der Baum vergrössert. Datenstrukturen, die sich so verändern können, nennt man *dynamische Datenstrukturen*.

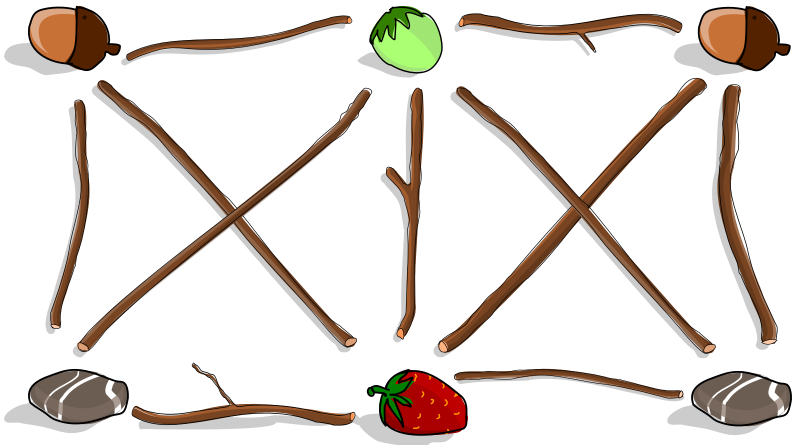
Stichwörter und Webseiten

- Binärer Suchbaum: `https://de.wikipedia.org/wiki/Binärer_Suchbaum`

7. Erdbeerklau

Anja will im Garten ein Kunstwerk schaffen und hat dafür verschiedene Sachen gesammelt: Mehrere Eicheln, Haselnüsse, Steine und eine Erdbeere. Sie legt einige der Sachen auf den Rasen.

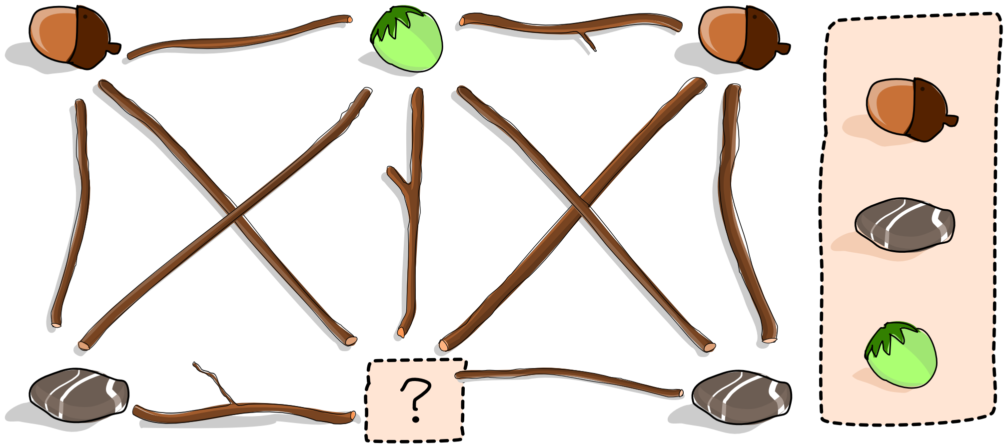
Danach legt Anja Äste zwischen diese Sachen. Dabei befolgt sie folgende Regel: Ein Ast darf nicht zwischen zwei gleichen Sachen liegen – zum Beispiel nicht zwischen zwei Eicheln. Hier ist das fertige Kunstwerk:



Während Anja weg ist, kommt ihr Bruder und isst die Erdbeere.

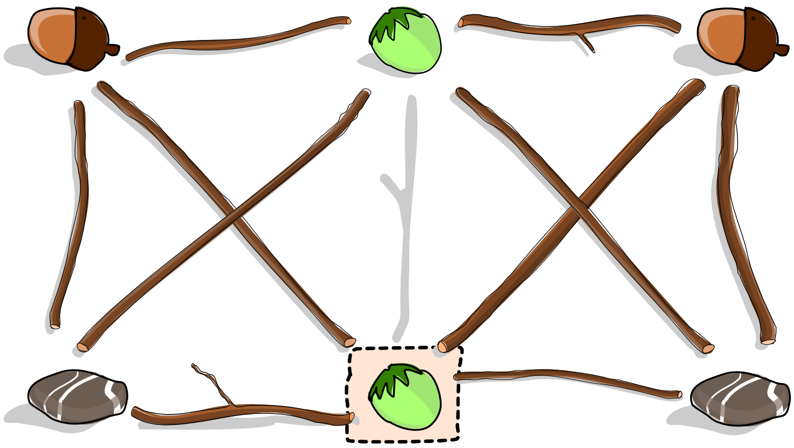
Kannst du ihm helfen, die Tat zu verschleiern?

Platziere eine andere Sache an die Stelle der Erdbeere und entferne genau einen Ast. Am Ende soll Anjas Regel auch für das veränderte Kunstwerk gelten.



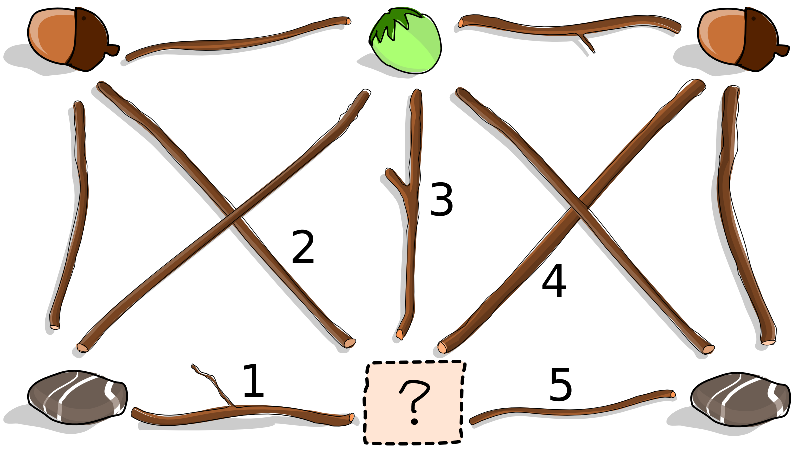
Lösung

Wenn man die Erdbeere durch eine Haselnuss ersetzt, verletzt der Ast 3 in der Mitte Anjas Regel: Er liegt zwischen zwei gleichen Sachen, nämlich zwei Haselnüssen. Deshalb muss dieser Ast entfernt werden.



Bei den beiden anderen möglichen Ersetzungen muss man mehr als einen Ast entfernen:

- Wird die Erdbeere durch eine Eichel ersetzt, muss man die Äste 2 und 4 entfernen.
- Wird die Erdbeere durch einen Stein ersetzt, muss man die Äste 1 und 5 entfernen.



Dies ist Informatik!

Anjas Kunstwerk kann als *Graph* dargestellt werden. Ein Graph besteht aus *Knoten* (den Plätzen für die Sachen) sowie aus *Kanten* (den Ästen), die jeweils zwei Knoten miteinander verbinden. Graphen sind sehr vielseitig und werden bei vielen Informatik-Aufgabenstellungen zur Modellierung verwendet. Wenn zwei Knoten direkt durch eine Kante verbunden sind, sind sie *Nachbarn* voneinander. Eine Gruppe von Knoten, in der jeder Knoten Nachbar von jedem anderen ist, heisst *Clique*. Im unserem Graphen haben wir zwei Cliquen mit vier Knoten: die rechte und die linke Hälfte des Graphen. (Die Haselnuss oben und das Fragezeichen gehören zu beiden Cliquen.) Aus Anjas Regel folgt, dass alle Knoten einer Clique mit unterschiedliche Sachen belegt sein müssen. Um die Regel einzuhalten, brauchen wir also mindestens so viele unterschiedliche Sachen wie eine Clique Knoten hat. Nachdem die Erdbeere entfernt wurde, haben wir aber nur noch 3 verschiedene Sachen. Also dürfen jetzt noch Cliquen mit höchstens 3 Knoten übrig bleiben, wenn die Regel weiterhin erfüllt bleiben soll. Es muss also eine Kante (ein Ast) entfernt werden, sodass beide Cliquen mit vier Knoten kaputt gehen.

Anjas Regel entspricht einer Regel im sogenannten *Färbungsproblem* für Graphen: Wir ordnen jedem Knoten eines Graphen eine Farbe zu, wobei Nachbarn unterschiedliche Farben haben müssen. (Die Farben entsprechen den verschiedenen Arten von Sachen.) Das Ziel ist meistens, möglichst wenig Farben zu benützen. Das Problem, wie man einen Graphen mit der minimalen Anzahl von Farben einfärbt, hat viele Anwendungen. Einige Beispiele sind die Planung von Sportwettkämpfen, das Entwerfen eines Sitzplans und sogar das Lösen eines Sudoku-Rätsels.

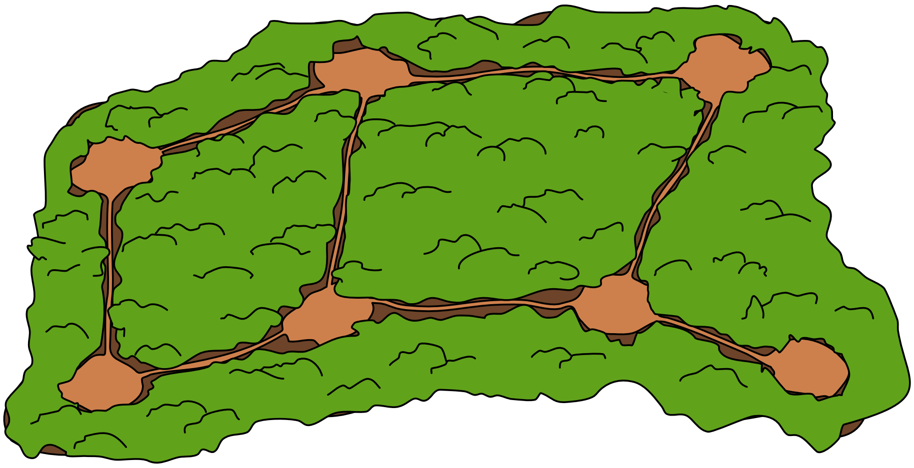
Stichwörter und Webseiten

- Färbeproblem: `https://de.wikipedia.org/wiki/Färbung_(Graphentheorie)`
- minimale Kantenfärbung, Kantenfärbung: `https://de.wikipedia.org/wiki/Kantenfärbung`
- Clique: `https://de.wikipedia.org/wiki/Clique_(Graphentheorie)`

8. Tiere beobachten

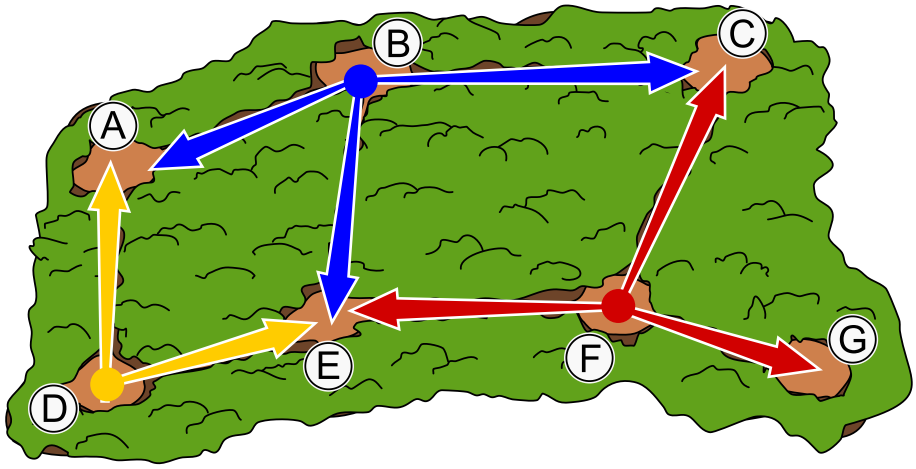
Die Förster wollen die Tiere auf den Wegen im Wald beobachten. Von jeder Lichtung aus können sie alle abgehenden Wege bis zur nächsten Lichtung beobachten. Es sollen möglichst wenig Förster alle Wege beobachten.

Wähle möglichst wenige Lichtungen, von denen die Förster alle Wege beobachten können!



Lösung

Das Bild zeigt die minimale Lösung, bei der Förster auf nur drei Lichtungen stehen und alle Wege beobachten können.



Es gibt acht Wege, die beobachtet werden müssen. Wenn nur zwei Förster alle Wege beobachten könnten, müsste es eine Lichtung geben, von der mindestens vier Wege fortführen. Aber eine solche Lichtung gibt es nicht in diesem Wald. Daher sind zwei Förster zu wenig.

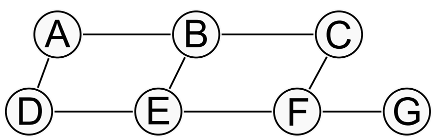
Es sind also mindestens drei Förster zur Beobachtung aller Wege notwendig. Daher ist die hier angegebene Lösung eine Lösung mit der kleinstmöglichen Anzahl an Förstern. Tatsächlich gibt es keine weitere Lösung mit genau drei Förstern.

Aus der Anzahl der zu überblickenden Wege und der Tatsache, dass es keine Lichtungen mit mehr als drei wegführenden Wegen gibt, können wir schliessen: Jeder Förster muss mindestens zwei Wege überblicken, die die anderen Förster nicht überblicken.

Um die Sackgasse zwischen Lichtung F und G zu überblicken, muss ein Förster auf der Lichtung F platziert werden. Um den Weg zwischen Lichtung B und C zu überblicken, muss der zweite Förster von der Lichtung B aus beobachten. Um die letzten zwei Wege mit nur einem Förster zu überblicken, muss dieser auf der Lichtung D platziert werden. Somit kommt man zwingend auf die angegebene Lösung und es kann keine weiteren geben.

Dies ist Informatik!

Beziehungen zwischen Dingen (z.B. Wege zwischen Lichtungen) können als sogenannter *Graph* dargestellt werden. Ein Graph besteht aus *Knoten* (hier: die Lichtungen), dargestellt als Kreise, und *Kanten* (hier: die Wege), dargestellt als Linien zwischen den Knoten. Der Graph dieser Aufgabe schaut so aus:



Bei dieser Biberaufgabe soll man in dem Graphen eine kleinste Anzahl an Knoten finden, sodass jede Kante bei mindestens einem dieser Knoten beginnt oder endet. So eine Teilmenge der Knoten nennen Informatiker eine *minimale Knotenüberdeckung* (engl. *minimal vertex cover*). Solche Knotenüberdeckungsprobleme finden wir im Alltag zum Beispiel beim Suchen der besten Stellen für Strassenlampen oder für das geschickte Platzieren von Überwachungskameras.

Stichwörter und Webseiten

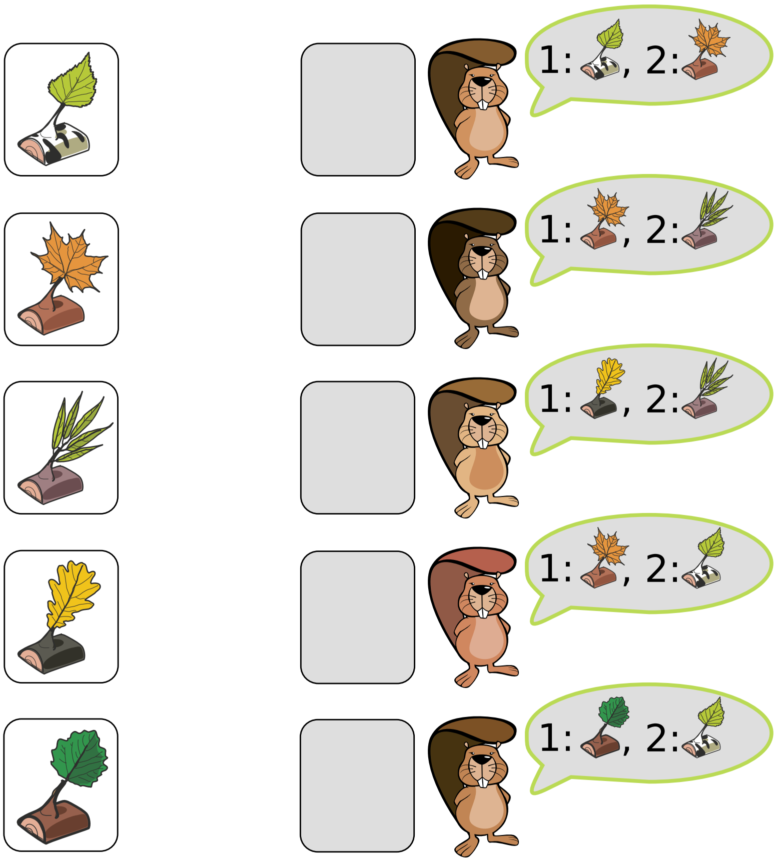
- Graph: `https://de.wikipedia.org/wiki/Graph_(Graphentheorie)`
- Knotenüberdeckung: `https://de.wikipedia.org/wiki/Knotenüberdeckung/`

9. Lieblingsgeschenk

Die Biberfamilie hat fünf Geschenke für ihre fünf Kinder. Jedes Kind nennt zuerst sein Lieblingsgeschenk und dann das zweitliebste. Die Geschenke sollen richtig zugeteilt werden:

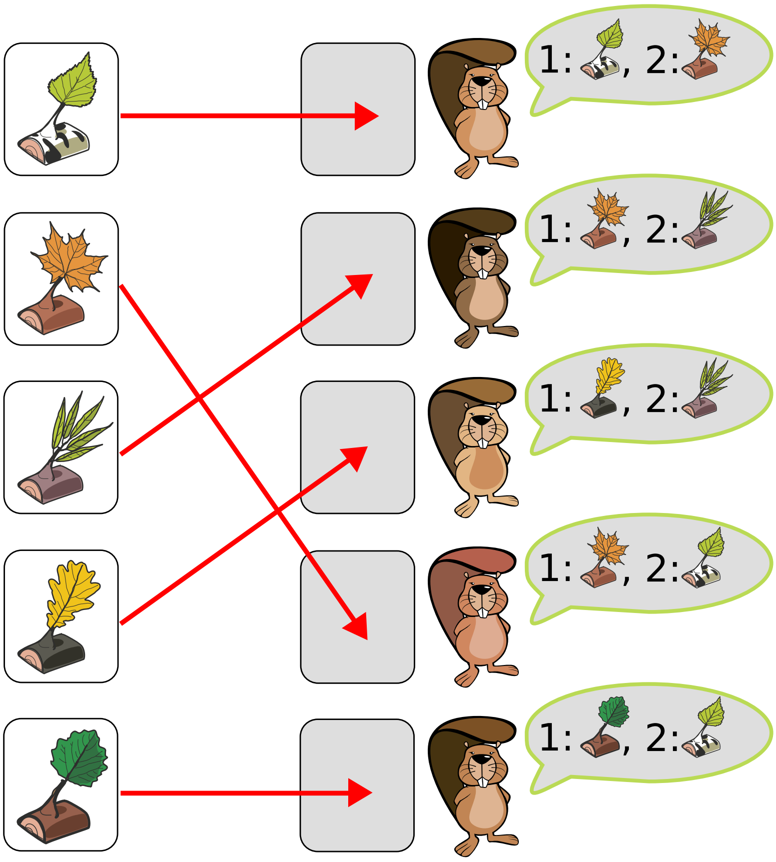
1. Möglichst viele Kinder sollen ihr Lieblingsgeschenk bekommen.
2. Die übrigen sollen das zweitliebste bekommen.

Gib den Kindern die richtigen Geschenke.



Lösung

Hier ist die einzige Zuordnung der Geschenke, die beide Bedingungen erfüllt.



Die obige Grafik ordnet vier Bibern ihr Lieblingsgeschenk zu und einem Biber sein zweitliebstes Geschenk. Nicht alle Biberkinder können ihr Lieblingsgeschenk bekommen, da zwei Biber dasselbe Lieblingsgeschenk haben. Daher ist keine Zuordnung möglich, bei der mehr Biber ihr Lieblingsgeschenk bekommen. Beachte: Wenn du die Zuteilung von oben nach unten vornimmst und das zweite Geschenk dem zweiten Biber zuteilst, dann bekommt der vierte Biber keines seiner bevorzugten Geschenke. In dieser Aufgabe genügt es also nicht, für jeden einzelnen Biber die jeweils beste aktuelle Auswahl zu treffen.

Eine Lösungsstrategie ist, dass du zuerst alle Geschenke zuordnest, die nur von einem Kind das Lieblingsgeschenk sind. Danach bleiben nur noch zwei Kinder mit demselben Lieblingsgeschenk übrig. Bei diesen schaust du, welchem das zweitliebste Geschenk zugeordnet werden kann. Dem anderem teilst du sein Lieblingsgeschenk zu.

Dies ist Informatik!

Bei dieser Aufgabe handelt sich um ein eindeutiges *Zuordnungsproblem*: Wir möchten die Geschenke so zuordnen, dass alle Kinder ein Geschenk bekommen und es kein Kind ohne Geschenk gibt. Dabei haben die Kinder nicht nur einen einzelnen Wunsch, sondern geben eine Reihenfolge von Vorlieben an. Solche Zuordnungsprobleme mit Reihenfolgen von Vorlieben können sehr kompliziert werden. Die Informatik hilft uns dabei, solche Probleme möglichst rasch zu lösen.

Eine Möglichkeit ist den Zuordnungen einen Wert zu geben: Das Lieblingsgeschenk hat Wert 1 und das zweitliebste Geschenk den Wert 2. Wir möchten den Gesamtwert minimieren. Eine *Zuordnung* (engl. *Matching*) ist *optimal*, wenn es keine andere Zuordnung mit mehr erfüllten Erstplatzierungen gibt. In der Informatik wird so eine Zuordnung als *Rang-Maximal-Matching* bezeichnet. Es gibt viele Matching-Probleme. Eines davon wird als *Problem der stabilen Paarung* (engl. *Stable Marriage Problem*) bezeichnet. Klingt interessant? Dann solltest du Informatik studieren!

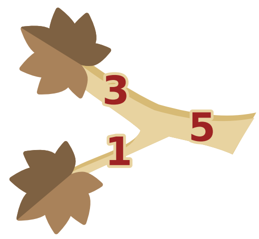
Stichwörter und Webseiten

- Zuordnungsproblem: `https://de.wikipedia.org/wiki/Zuordnungsproblem`
- Rang-Wahl: `https://de.wikipedia.org/wiki/Rang-Wahl`

10. Rette den Baum!

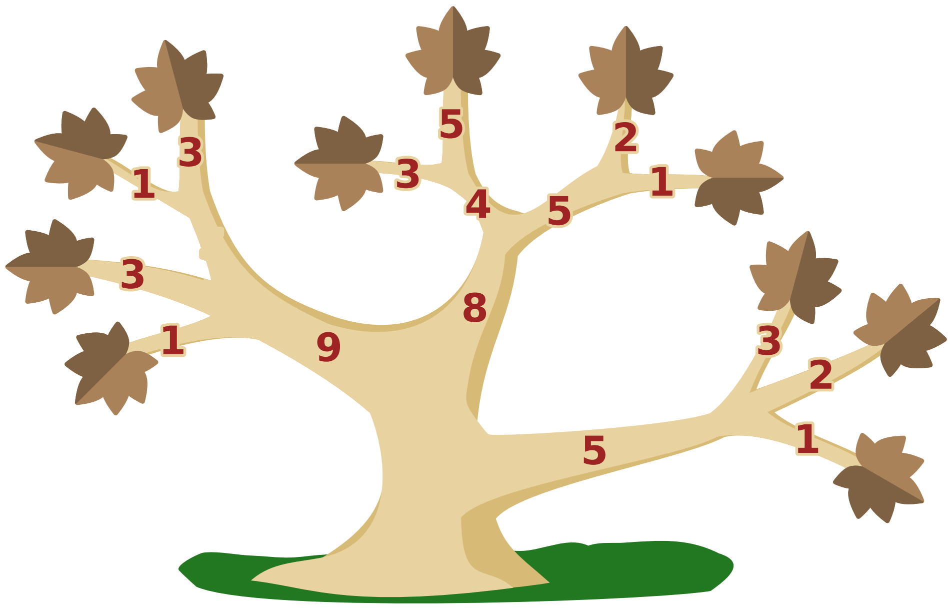
Ein Baum in Brunos Garten ist krank, alle Blätter sind vertrocknet. Bruno will den Baum retten. Dazu muss er einige Äste absägen, so dass am Ende alle Blätter entfernt sind. Dann können neue Äste mit neuen Blättern wachsen. Bruno möchte so schnell wie möglich fertig sein.

Das Bild zeigt ein Beispiel:



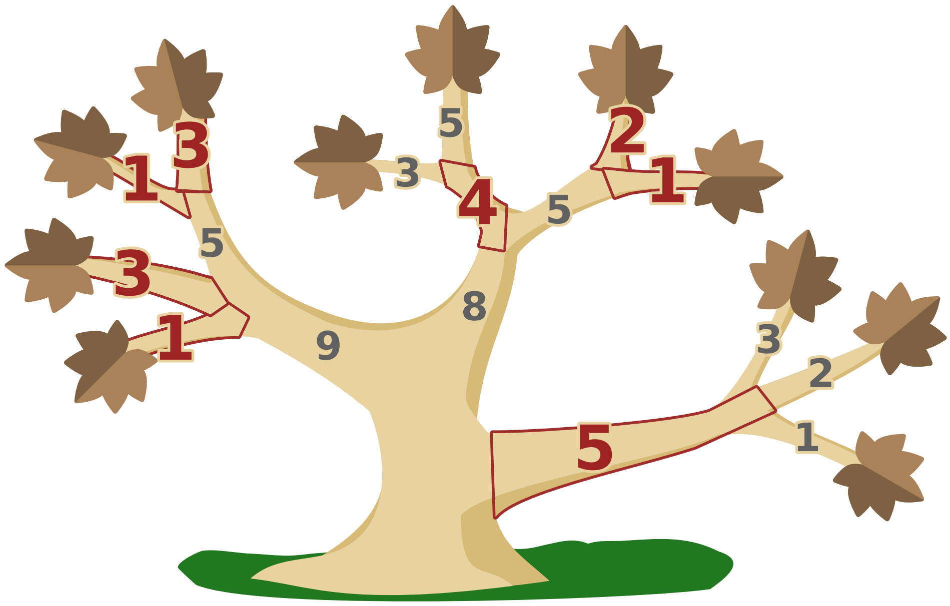
Um die beiden Blätter zu entfernen, kann Bruno entweder die beiden Äste mit den Blättern absägen oder nur den einen Ast, von dem die beiden anderen abzweigen. Die Zahlen geben für jeden Ast an, wie lange das Absägen dauert. Bruno wird also die beiden Äste mit den Blättern absägen, da $3 + 1 < 5$. Unten siehst du den gesamten Baum.

Welche Äste wird Bruno absägen, um so schnell wie möglich fertig zu sein?



Lösung

So ist es richtig: Bruno sägt die rot markierten Äste ab, um so schnell wie möglich fertig zu sein:



Aber warum ist das so? Zunächst können wir ausrechnen, wie viel Zeit Bruno benötigt, wenn er nur die Äste mit Blättern absägt – damit wäre er ja fertig:

$$1 + 3 + 1 + 3 + 3 + 5 + 2 + 1 + 3 + 2 + 1 = 25$$

Nun gehen wir in Richtung Stamm weiter und überlegen immer wieder neu, ob es schneller sein könnte, den Ast abzusägen, von dem die bisherigen Äste direkt oder indirekt abzweigen. Nach dem ersten solchen Schritt ergibt sich die folgende Rechnung (die Funktion «min» berechnet das Minimum ihrer Argumente):

$$\begin{aligned} & 1 + 3 + \min(5, 1 + 3) + \min(4, 3 + 5) + \min(5, 2 + 1) + \min(5, 3 + 2 + 1) \\ = & 1 + 3 + (1 + 3) + 4 + (2 + 1) + 5 \\ = & 20 \end{aligned}$$

Dabei rechnen wir die Gesamtzeit zuerst noch nicht aus, damit wir besser sehen, welche Äste abzusägen sind. Nach dem nächsten Schritt sind wir schon am Stamm angekommen:

$$\begin{aligned} & \min(9, 1 + 3 + 1 + 3) + \min(8, 4 + 2 + 1) + 5 \\ = & (1 + 3 + 1 + 3) + (4 + 2 + 1) + 5 \\ = & 20 \end{aligned}$$

Schneller kann Bruno nicht fertig sein.

Dies ist Informatik!

Stellen wir uns einmal vor, die abgesägten Stücke von Brunos Baum würden nicht direkt auf die Erde fallen – so wie beim Lösen dieser Biberaufgabe am Bildschirm. Dann könnte man sagen, dass der Baum durch das Absägen in nur zwei Teile zerlegt wird: Der eine Teil enthält alle abgesägten Stücke, insbesondere also alle Blätter, und der andere Teil enthält den Stamm und alle davon ausgehenden Äste bis zu den Sägestellen. Diese Zweiteilung bzw. dieser *Schnitt* durch den Baum ist minimal in Bezug auf die Zeit, die Bruno fürs Absägen aufbringen muss.

Auch die Informatik kennt Bäume und verwendet sie zur Darstellung von Objekten, die auf bestimmte Weise miteinander verbunden sind. Die Objekte werden als *Knoten* bezeichnet, die Verbindungen als *Kanten*. Dabei gibt es zwischen zwei Knoten immer nur einen Weg entlang der Kanten – so wie es im echten Baum von einem Blatt oder einer Astgabelung immer nur einen Weg entlang der Äste zum Stamm gibt. Verzichtet man auf diese Bedingung, ist allgemeiner von einem *Graphen* die Rede.

In einem allgemeinen Graphen ist ein *minimaler Schnitt*, also die Zerlegung in zwei oder auch mehrere Teile mit minimalen Kosten, nicht so einfach zu berechnen, wie wir es hier für einen Baum vorgemacht haben, aber auch nicht allzu schwierig. Das ist gut, denn es gibt interessante Anwendungen. Minimale Schnitte können etwa bei der Zerlegung von Bilddateien in ähnliche Teile verwendet werden. In speziellen Graphen, den *Flussnetzen*, mit denen unter anderem Datenflüsse durch Netzwerke modelliert werden können, entsprechen die Kosten eines minimalen Schnitts dem maximal möglichen Fluss durch das gesamte Netz.

Stichwörter und Webseiten

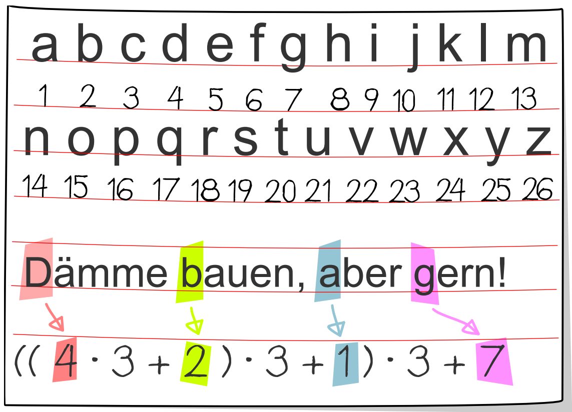
- Baum: `https://de.wikipedia.org/wiki/Baum_(Graphentheorie)`
- Schnitt: `https://de.wikipedia.org/wiki/Schnitt_(Graphentheorie)`
- Flüsse und Schnitte in Netzwerken:
  `https://de.wikipedia.org/wiki/Flüsse_und_Schnitte_in_Netzwerken`
- Max-Flow-Min-Cut-Theorem:
  `https://de.wikipedia.org/wiki/Max-Flow-Min-Cut-Theorem`

11. Bibliothek

Susi ist mit Tim in der Biber-Bibliothek. Sie wollen ein Buch ausleihen: «Dämme bauen, aber gern!»

Tim geht zu Regal 1, greift in Reihe 3, Fach 6 und holt das Buch heraus. Susi ist beeindruckt. Tim erklärt Susi, wie man den Ort eines Buches bestimmt:

Man nimmt von jedem Wort im Titel den Anfangsbuchstaben und bestimmt seine Position im Alphabet. Nach und nach werden diese Positionen addiert, aber vor jedem Addieren wird der bisher erreichte Wert mit 3 multipliziert. Für das gewünschte Buch ergibt sich 136. Schon ist klar, wo das Buch steht.



Nun stellt Susi für ihre Lieblingsbücher die entsprechenden Rechnungen auf. In einem Fall hat sie aber einen Fehler gemacht.

In welchem?

A) Gutes gegen Biber-Fieber
$((7 \cdot 3 + 7) \cdot 3 + 2) \cdot 3 + 6$

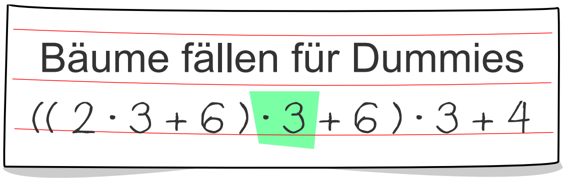
B) Bäume fällen für Dummies
$((2 \cdot 3 + 6) + 6) \cdot 3 + 4$

C) Der Herr der Dämme
$((4 \cdot 3 + 8) \cdot 3 + 4) \cdot 3 + 4$

D) Bebretti: Der erste Fall
$((2 \cdot 3 + 4) \cdot 3 + 5) \cdot 3 + 6$

Lösung

Susi hat fast alles richtig gemacht: Sie hat immer die richtigen Positionswerte addiert, und sie hat die Zwischenergebnisse immer mit 3 multipliziert – mit einer Ausnahme: In Antwort B hat sie Letzteres einmal vergessen.



Dies ist Informatik!

Mit den «Orts-Bestimmungs-Ausdrücken» ermöglicht die Bibliothek ihren Besuchern, die Standorte der Bücher genau zu bestimmen. So muss niemand lange suchen. Eines müssen die Bibliothek und Besucher aber beachten: Für verschiedene Bücher können die Ausdrücke und damit auch deren Ergebnisse gleich sein. Zum Beispiel stehen «Bäume fällen für Dummies» und «Biber finden Fichten dufte» im gleichen Fach. Die Fächer dürfen also nicht zu klein sein, oder sie müssen flexibel angepasst werden können.

Auch bei Daten, die in Computerspeichern abgelegt werden, ist es eine gute Idee, wenn ihr Speicherort direkt aus den Daten selbst berechnet werden kann. Dafür wurden in der Informatik *Hash-Funktionen* entwickelt: mathematische Funktionen, die aus dem Inhalt der Daten bzw. eines Teils der Daten einen Wert berechnen, der direkt den Speicherort angibt – so wie bei den Buchtiteln in dieser Biberaufgabe. Gute Hash-Funktionen sorgen dafür, dass sich in möglichst wenigen Fällen der gleiche Wert ergibt. Kommt eine solche Kollision doch einmal vor, kennt die Informatik gute Methoden, damit umzugehen.

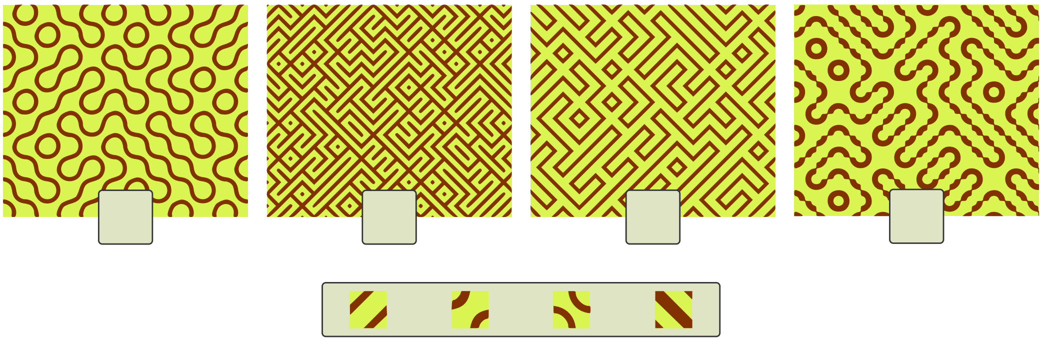
Stichwörter und Webseiten

- Hashfunktion: `https://de.wikipedia.org/wiki/Hashfunktion`
- Hashtabelle: `https://de.wikipedia.org/wiki/Hashtabelle`
- `http://www.abenteuer-informatik.de/PDF/ai2020_oa_leseversion.pdf`, Kapitel 11: Ordnung im Chaos

12. Fliesenmuster

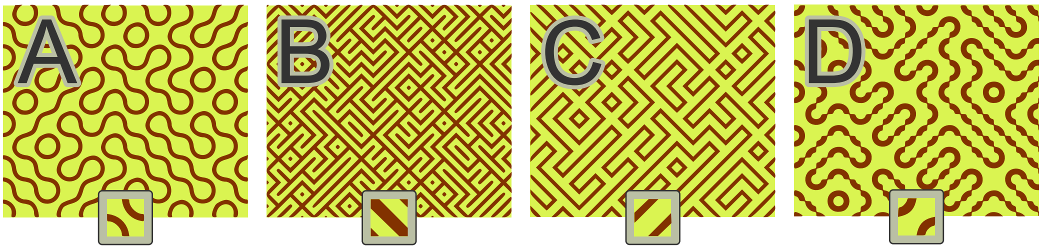
Die folgenden Muster wurde jeweils durch eine einzelne Fliese erzeugt. Die einzelnen Fliesen sind vergrössert dargestellt.

Ordne die Fliesen ihren möglichen Mustern zu.

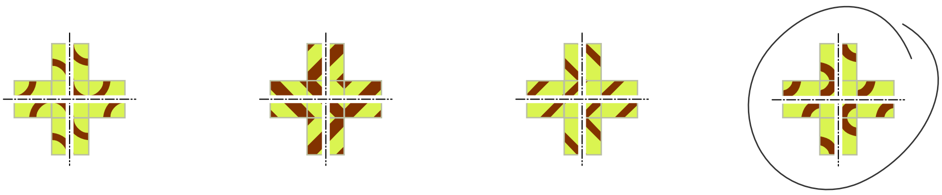


Lösung

Das ist die richtige Zuordnung:



Legt man jeweils 5 Fliesen aneinander und vergleicht sie genauer, erkennt man deutliche Unterschiede:



Fliese  hat als einzige der Fliesen vier Seiten, die nicht genau zueinander passen. Nur dadurch können Linien mit veränderlicher Breite wie in Muster D entstehen. Fliese  ist die einzige, die quadratischen Punkte in Muster B erzeugen kann, nämlich mit jeweils vier zusammenstossenden Dreiecken. Zudem hat sie den grössten Flächenanteil von Braun gegenüber Gelb, genau wie B; auch daran kann man die Zusammengehörigkeit erkennen. Somit verbleibt für die runden Formen von Fliese  nur Muster A als mögliches Ergebnis und für die geraden Formen von Fliese  nur Muster C.

Dies ist Informatik!

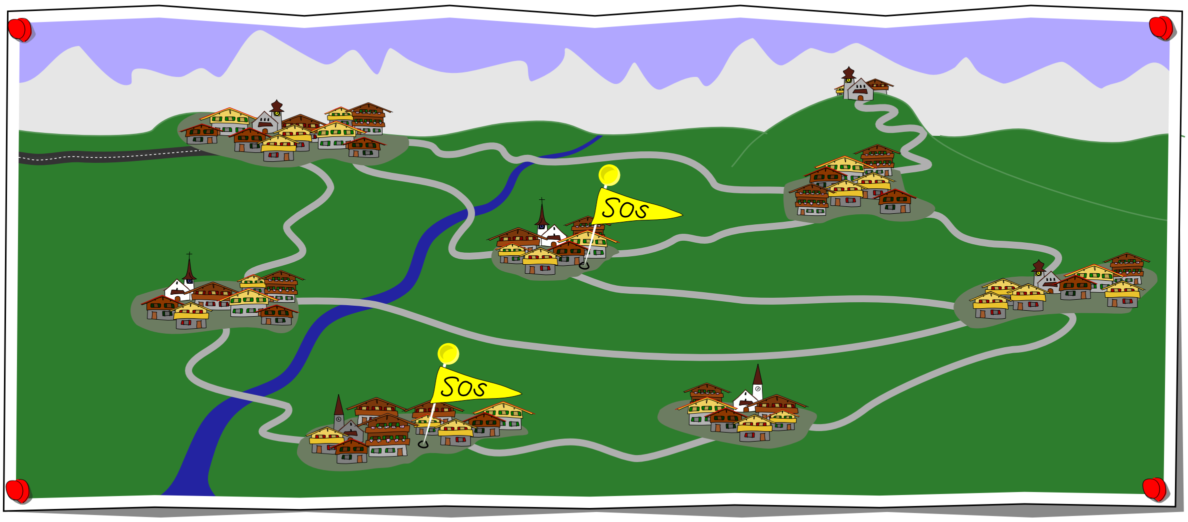
Diese Fliesen sind nach Sébastien Truchet (* 1657; † 1729) benannt, der an verschiedenen Varianten dieser Fliesen gearbeitet hat. Kacheln mit 4 gleichen Seiten bilden eine Untermenge von Truchet-Kacheln (Truchet-Kacheln müssen aber nicht unbedingt 4 gleiche Seiten haben, wie in 3 von den Mustern oben). Dass mit ganz einfachen Bausteinen komplexe Muster erstellt werden können, ist eine interessante Eigenschaft, die uns in der Informatik immer wieder begegnet. Truchet-Kacheln werden in der Mathematik und der Informatik untersucht und in Computerspielen verwendet, um Labyrinthe oder Dekorationen zu erstellen.

Stichwörter und Webseiten

- Truchet-Kacheln: `https://en.wikipedia.org/wiki/Truchet_tiles`

13. SOS aus den Bergen

Einige Bergdörfer werden aus der grossen Stadt über folgendes Strassennetz versorgt:

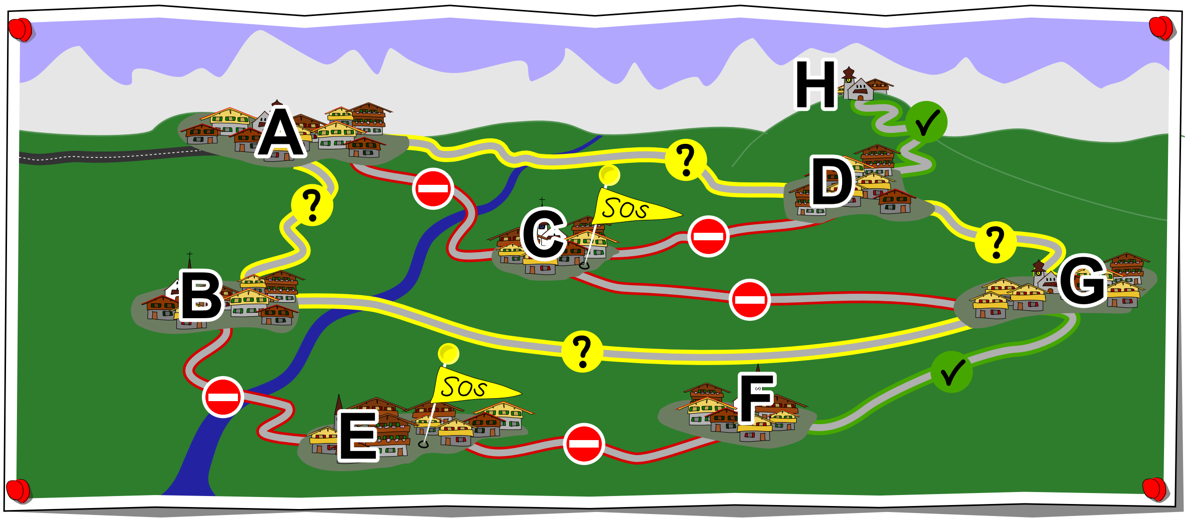


Nach einem Unwetter melden mehrere Dörfer, dass diese nicht mehr erreichbar sind, nämlich jene mit den SOS-Markierungen. Wir können daraus schliessen, dass einige Strassen blockiert sind.

Gib für jede Strasse zwischen den Dörfern in diesem Strassennetz an, ob diese (1) blockiert ist ⛔, (2) befahrbar ist ✅, oder (3) ob wir nicht ohne weitere Informationen sagen können, ob die Straße befahrbar oder blockiert ist ❓.

Lösung

Die Karte zeigt, was wir über die Verbindungen im Strassennetz wissen:



Wir beginnen mit dem Aufspüren der blockierten Strassen. Die zwei Strassen, die zu Dorf E führen, können blockiert sein, da ansonsten Dorf E noch erreichbar wäre. Ebenso sind die drei Strassen zu Dorf C blockiert, da sonst Dorf C noch erreichbar wäre.

Als nächstes suchen wir die Strassen, die befahrbar sein müssen. Die Strasse zwischen Dorf G und F muss befahrbar sein, da ansonsten, aufgrund der blockierten Strasse zwischen Dorf F und E, das Dorf F nicht erreichbar wäre. Auch die Strasse zwischen der Kirche H und dem Dorf D muss befahrbar sein, da H erreichbar ist und nur über D erreicht werden kann.

Nun bleiben die nur möglicherweise befahrbaren Strassen übrig. Da die Dörfer B, G und D mehrfach mit dem Dorf A verbunden sind, können wir nicht sagen können, welche der verbleibenden Strassen befahrbar sind. So könnte das Dorf B beispielsweise über Dorf A, aber auch Dorf G erreicht werden. Dasselbe gilt auch für Dorf D. Das Dorf G kann entweder über Dorf B oder D versorgt werden. Irgendeine der Strassen im Kreislauf A – B – G – D – A könnte also blockiert sein und diese 4 Dörfer könnten trotzdem alle erreichbar bleiben.

Dies ist Informatik!

So wie in Strassennetzen können auch bei Computernetzwerken Verbindungen fehlerhaft, überlastet oder ganz defekt sein. Um Ausfälle zu verhindern, werden oft Sicherheitsmassnahmen, wie z. B. mehrere Verbindungen zu einem Ort, eingeplant. Dies nennt man *Redundanz*.

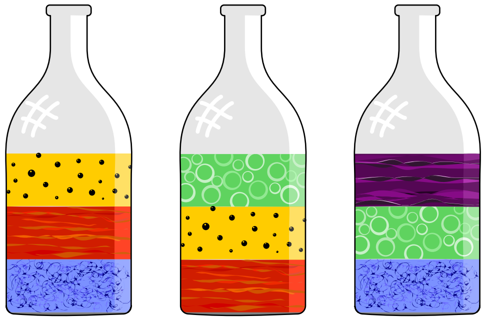
Das Beheben von Fehlern in einem System ist eine Aufgabe, die Informatiker sehr oft erledigen müssen, nicht nur in Computernetzwerken, sondern auch in der Softwareentwicklung. Um einen Fehler zu beheben, muss man seine genaue Quelle identifizieren, und dieser Prozess erfolgt meist schrittweise in mehreren Schritten. Einige Programmierer glauben, dass man nie alle Fehler und Bugs in einem Programm finden kann.

Stichwörter und Webseiten

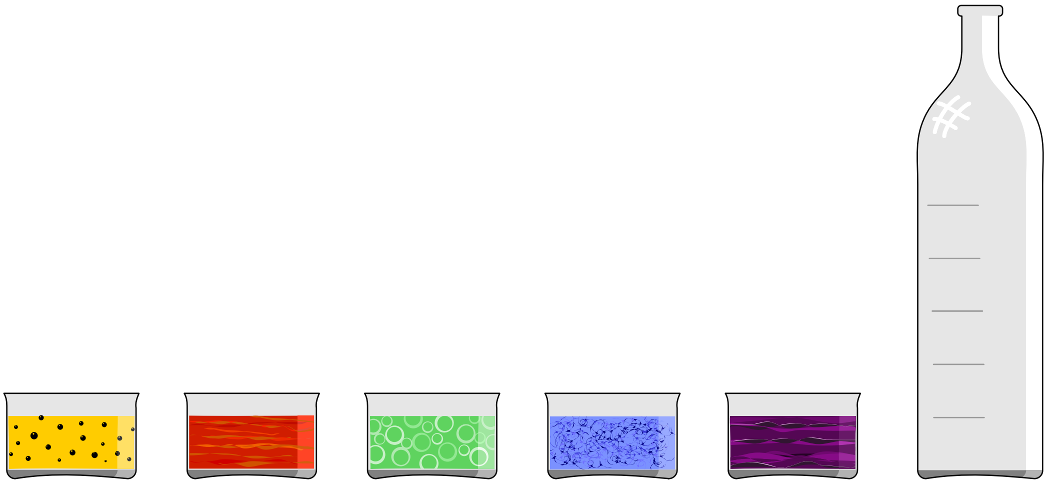
- Redundanz: `https://de.wikipedia.org/wiki/Redundanz_(Technik)`
- Debuggen: `https://de.wikipedia.org/wiki/Debuggen`

14. Schichte nach Dichte!

Mark hat Flaschen mit jeweils drei farbigen Flüssigkeiten, die übereinander geschichtet sind. Er weiss, dass sich die Flüssigkeiten mit geringerer Dichte immer über Flüssigkeiten mit grösserer Dichte bewegen. Nun möchte er sehen, wie es aussieht, wenn man alle farbigen Flüssigkeiten in eine Flasche gibt.

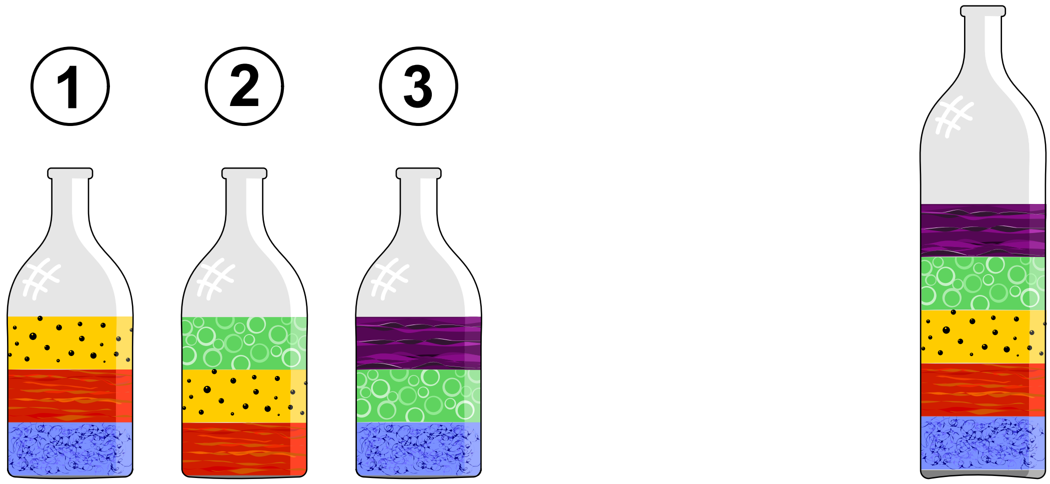


Ordne die fünf farbigen Flüssigkeiten in der Flasche, so wie sie am Ende angeordnet sind!



Lösung

Das Bild zeigt die richtige Anordnung der fünf farbigen Flüssigkeiten in der grossen Flasche.



Du findest die Reihenfolge nach folgendem Verfahren: Schritt für Schritt entfernst du in Gedanken aus den drei gegebenen Flaschen die Flüssigkeiten, die nicht auf einer anderen Flüssigkeit liegen, und gibst sie in die grosse Flasche.

Zu Beginn hat nur in den beiden Flaschen 1 und 3 blaue Flüssigkeit und dort ist sie ganz unten, sie liegt also nirgendwo auf einer anderen Flüssigkeitsschicht. Die rote Flüssigkeit ist zwar in Flasche 2 ganz unten. Aber in Flasche 1 liegt sie auf der blauen Flüssigkeit und muss deshalb eine geringere Dichte als die blaue haben. Also wird als erstes die blaue Flüssigkeit aus den Flaschen entfernt und in die grosse Flasche gegeben.

Jetzt ist die rote Flüssigkeit die einzige, die nicht auf einer anderen Flüssigkeit liegt. Sie wird aus den Flaschen 1 und 2 entfernt und in die grosse Flasche gegeben. Danach kommt die gelbe, dann die grüne und zum Schluss die violette Flüssigkeit, die die geringste Dichte hat und über der keine andere Flüssigkeit liegt.

Dies ist Informatik!

Bei der Lösung dieser Aufgabe hast du die Anordnung der Flüssigkeiten in den drei Flaschen der Aufgabenstellung ausgewertet und die Flüssigkeiten nach ihrer Dichte sortiert.

Ein Stoff hat viele messbare Eigenschaften: Siedetemperatur, Schmelztemperatur, elektrische Leitfähigkeit und eben die Dichte. In diesem Fall wurde die Dichte als Kriterium verwendet, Stoffe zu sortieren.

In vielen Computerprogrammen spielt das Sortieren von Daten eine wichtige Rolle. Das Verfahren, das bei dieser Aufgabe zur Ermittlung der Reihenfolge der Flüssigkeitsschichten verwendet wurde, nennt man *topologisches Sortieren*. Es wird zum Sortieren von Objekten angewendet, für die Beziehungen der Art Vorgänger/Nachfolger bekannt sind.

Stichwörter und Webseiten

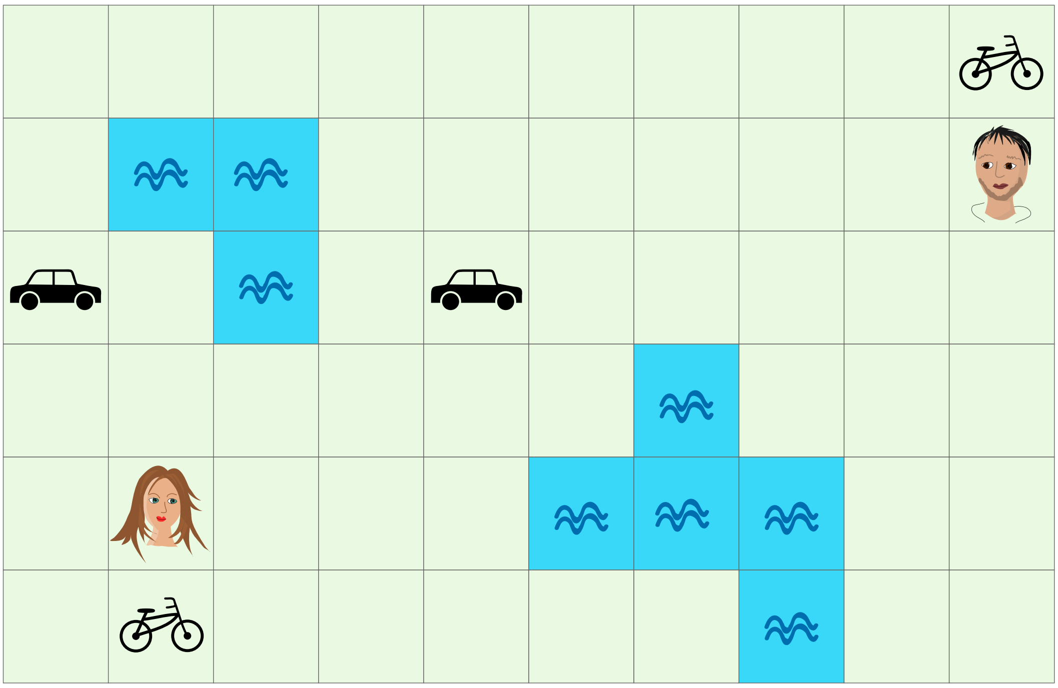
- Sortieren, Ordnung: `https://de.wikipedia.org/wiki/Sortierung`
- Topologisches Sortieren: `https://de.wikipedia.org/wiki/Topologische_Sortierung`, `https://www.ac.tuwien.ac.at/files/teaching/ss12/AD1/top_sortieren.pdf`

15. Es pressiert!

Zwei Freunde wollen sich möglichst bald treffen. Von einem Feld können sie sich zu einem benachbarten Feld links, rechts, oben oder unten bewegen.

Zu Fuss benötigen sie dafür 1 Minute. Wenn sie auf ein Feld mit einem Fahrzeug gelangen, können sie es benutzen.

Mit einem Fahrrad schaffen sie in einer Minute 2 Felder und mit einem Auto 5 Felder. Dabei sind Richtungsänderungen möglich. Wasserflächen können sie nicht überqueren.

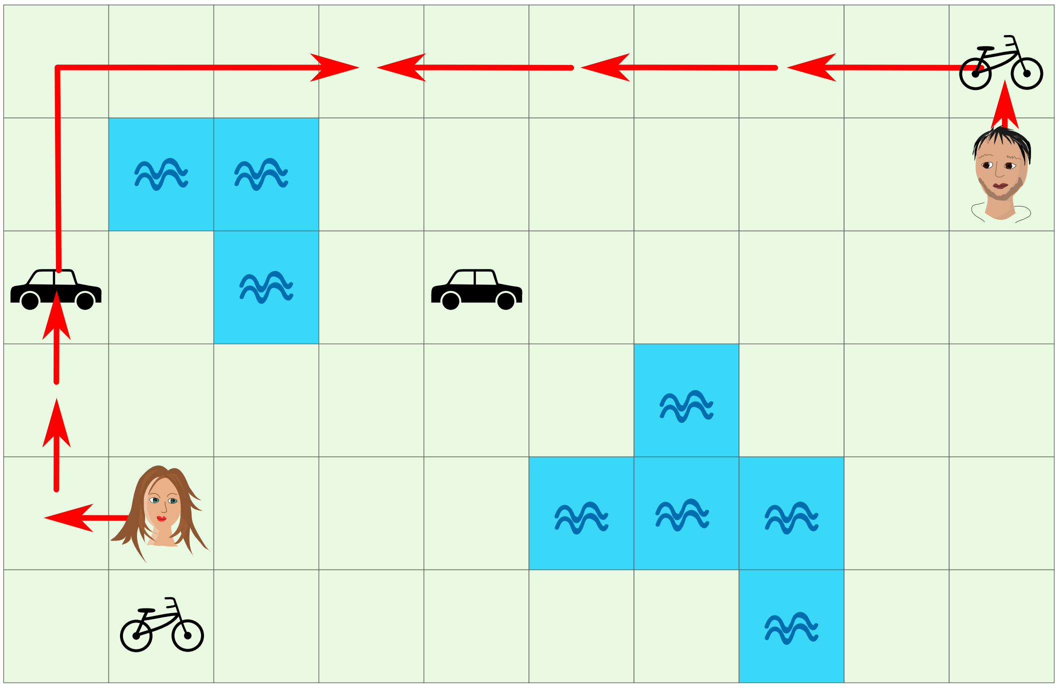


Wie viele Minuten benötigen die beiden Freunde mindestens, um sich auf einem Feld zu treffen?

A) 1 Minute
B) 2 Minuten
C) 3 Minuten
D) 4 Minuten
E) 5 Minuten
F) 6 Minuten

Lösung

Die richtige Antwort ist D) 4 Minuten. Das Bild zeigt eine Route, mit der sich die beiden Freunde in 4 Minuten auf einem Feld treffen können.



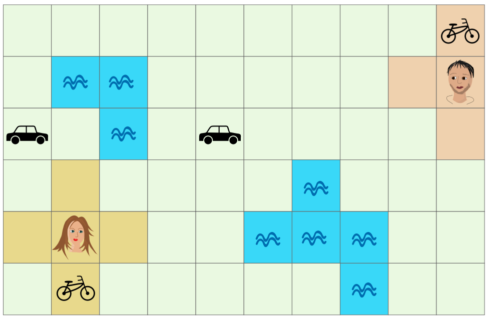
Nun muss noch bewiesen werden, dass sie sich nicht in 3 Minuten treffen können: Die beiden Freunde sind 11 Felder voneinander entfernt. In 3 Minuten können sie aber zu Fuss insgesamt nur 6Felder näher zueinander kommen. Wenn einer das Fahrad erreicht hat und der andere zu Fuss geht, dann können sie in 3 Minuten 9 Felder näher zueinander kommen, was auch nicht reicht. Selbst wenn beide zu einem Fahrrad gehen, reicht es nicht. Denn dann könnten sie in 3 Minuten 12 Felder näher zueinander kommen, die beiden Fahrräder sind aber 13 Felder voneinander entfernt.

Also bleibt nur die Option, ein Auto zu benützen. In 3 Minuten kann nur das Mädchen ein Auto erreichen. Es bleibt dann aber keine Zeit mehr, das Auto zu benutzen. Und in 3 Minuten kann der Junge kein Feld mit einem Auto erreichen.

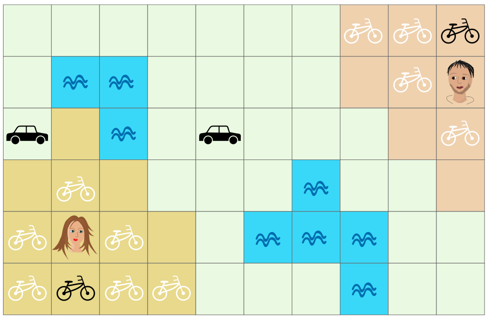
Dies ist Informatik!

Wie hast du die Aufgabe gelöst? Hast du zufällig eine schnelle Route gefunden und gehofft, dass es keine schnellere gibt? Oder hast du viele Möglichkeiten ausprobiert und dir die schnellste gemerkt?

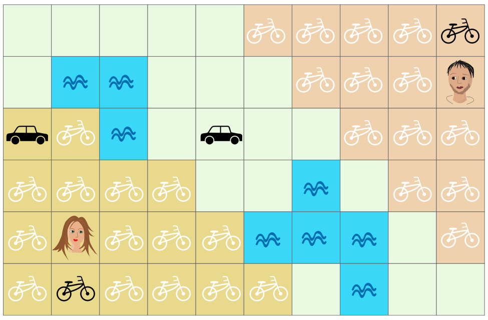
Computerprogramme, die für diese Art von Problemen entwickelt worden sind, arbeiten meist nach einem Verfahren, das man *Breitensuche* nennt. Bei dieser Aufgabe geht die Breitensuche folgendermassen:



1. Markiere alle Felder, die von den beiden Freunden in einer Minute erreicht werden können.

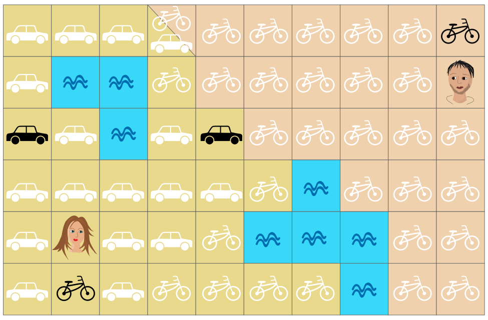


2. Markiere alle Felder, die in (höchstens) einer Minute von den Feldern aus erreicht werden können, die in Schritt 1 markiert worden sind. Notiere auch das verwendete Verkehrsmittel.



3. Markiere alle Felder, die in einer Minute von den Feldern aus erreicht werden können, die in Schritt 2 markiert worden sind.

Weil die beiden Gebiete, die du bisher markiert hast, sich nicht überlappen, können sich die Freunde nach drei Minuten noch nicht treffen.



4. Markiere nun alle Felder, die in einer Minute von den in Schritt 3 markierten Feldern erreicht werden können.

Nun überlappen sich die beiden Gebiete in einem Feld. Es kann nach 4 Minuten von dem Mädchen mit einem Auto und von dem Jungen mit einem Fahrrad erreicht werden. Navigationssysteme finden den schnellsten Weg zwischen zwei Punkten. Sie achten dabei darauf, dass die Route über geeignete Strassen und Wege verläuft - und nicht etwa querfeldein und durch Flüsse. Diese Aufgabe ähnelt dem Navigationsproblem, nur müssen hier zwei Personen zu einem gemeinsamen – anfangs noch unbekannten - Ziel geführt werden und nicht nur eine Person zu einem festen Ziel.

Weil ein Computer bei der Breitensuche systematisch vorgeht, findet er auch Lösungen, die nicht direkt ins Auge springen. Manchmal ist ein Umweg mit weniger Ampeln schneller, als die kürzeste Route zwischen Start und Ziel. Eine Bahnverbindung mit Umsteigen kann schneller sein als eine direkte Busverbindung. In der Informatik kennt man mehrere Verfahren, um die beste Lösung zu einem Problem dieser Art zu finden. Abgesehen von der Breitensuche, die gerade beschrieben worden ist, gibt es auch einen Ansatz, den man *Branch and Bound* nennt (engl. für *Verzweigen und Begrenzen*).

Bei der Breitensuche wird jede Teillösung, die mit einer bestimmten Anzahl von Arbeitsschritten erreicht wird, berücksichtigt. Bei *Branch and Bound* verfolgt man Teillösungen nicht weiter, wenn man weiss, dass sie nicht zur optimalen Lösung führen können.

Wenn ein Problem zu komplex wird, würde es auch für den schnellsten Computer der Welt zu lange dauern, alle Möglichkeiten durchzuspielen, um die beste Lösung zu finden. In der Praxis reicht es bei einem Navigationssystem häufig aus, eine sehr gute Route zu finden, auch wenn es nicht die bestmögliche ist. (Wenn du dein Ziel in 78 Minuten erreichen kannst, macht es dir vermutlich nichts aus, wenn man es theoretisch auch in 77 Minuten erreichen könnte.)

Stichwörter und Webseiten

- Breitensuche: `https://de.wikipedia.org/wiki/Breitensuche`
- Branch and Bound Algorithmus: `https://de.wikipedia.org/wiki/Branch-and-Bound`

A. Aufgabenautoren

Sarah Atkins
Michael Barot
Liam Baumann
Wilfried Baumann
Javier Bilbao
Sarah Chan
Kris Coolsaet
Valentina Dagienė
Christian Datzko
Susanne Datzko
Margarita Flores-Sicich
Fabian Frei
Gerald Futschek
Thomas Galler
Yasemin Gülbahar
Ezgi Arzu Güneş
Juraj Hromkovič
Alisher Ikramov
YongJu Jeon
Soojin Jun
Filiz Kalelioğlu
Dong Yoon Kim
Jihye Kim
Vaidotas Kinčius
Regula Lacher
Taina Lehtimäki
Marielle Léonard
Tom Naughton
Mochammad Irfan Noviana
Jean-Philippe Pellet
Zsuzsa Pluhár
Wolfgang Pohl
Peter Rossmanith
Eljakim Schrijvers
Tomas Šiaulys
Timur Sitdikov
Bernadette Spieler
Ahto Truu
Florentina Voboril
Michael Weigend
Kyra Willekes
Hongjin Yeh
Mija Zaļūksne

B. Sponsoring: Wettbewerb 2021

`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.

`http://www.baerli-biber.ch/`
Schon in der vierten Generation stellt die Familie Bischofberger ihre Appenzeller Köstlichkeiten her. Und die Devise der Bischofbergers ist dabei stets dieselbe geblieben: «Hausgemacht schmeckt's am besten». Es werden nur hochwertige Rohstoffe verwendet: reiner Bienenhonig und Mandeln allererster Güte. Darum ist der Informatik-Biber ein «echtes Biberli».

`http://www.verkehrshaus.ch/`

Standortförderung beim Amt für Wirtschaft und Arbeit Kanton Zürich

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://www.ubs.com/`
Wealth Management IT and UBS Switzerland IT

`http://www.oxocard.ch/`
OXOcard: Spielend programmieren lernen
OXON

`https://educatec.ch/`
educaTEC
Wir sind MINT-Experten. Seit unserer Gründung 2004 verfolgen wir das Ziel, Technik und ingenieurwissenschaftliches Denken in öffentlichen und privaten Schulen der Schweiz zu fördern. In Kombination mit kompetenter Beratung und Unterstützung offerieren wir Lehrkräften innovative Lehrmaterialien von weltweit führenden Herstellern sowie Lernkonzepte für den MINT-Bereich und verwandte Fächer.

`http://senarclens.com/`
Senarclens Leu & Partner

`http://www.abz.inf.ethz.ch/`
Ausbildungs- und Beratungszentrum für Informatikunterricht der ETH Zürich.

`http://www.hepl.ch/`
Haute école pédagogique du canton de Vaud

`http://www.phlu.ch/`
Pädagogische Hochschule Luzern

`https://www.fhnw.ch/de/die-fhnw/hochschulen/ph`
Pädagogische Hochschule FHNW

`http://www.supsi.ch/home/supsi.html`
La Scuola universitaria professionale della Svizzera italiana (SUPSI)

https://www.phzh.ch/
Pädagogische Hochschule Zürich

C. Weiterführende Angebote

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.

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.

Werden Sie SVIA Mitglied – `http://svia-ssie-ssii.ch/svia/mitgliedschaft` und unterstützten 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.