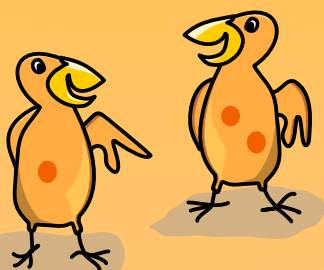




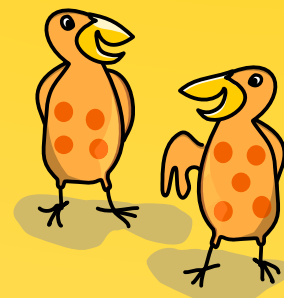
INFORMATIK-BIBER SCHWEIZ
CASTOR INFORMATIQUE SUISSE
CASTORO INFORMATICO SVIZZERA

Aufgaben und Lösungen 2021

Schuljahre 5/6



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



Herausgeber:

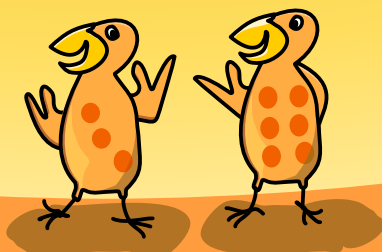
Susanne Datzko, Fabian Frei,
Jean-Philippe Pellet



010100110101011001001001
010000010010110101010011
010100110100100101000101
001011010101001101010011
010010010100100100100001

SV!A

www.svia-ssie-ssii.ch
schweizerischerverein für informatik in d
erausbildung // société suisse pour l'infor
matique dans l'enseignement // società sviz
zera per l'informatica nell'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 Rossmann: ETH Zürich, Ausbildungs- 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.



Die Aufgaben sind lizenziert unter einer Creative Commons Namensnennung – Nicht-kommerziell – Weitergabe unter gleichen Bedingungen 4.0 International Lizenz. Die Autoren sind auf S. 42 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 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.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 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

Mitarbeit Informatik-Biber 2021	i
Vorwort	iii
Inhaltsverzeichnis	v
1. Brückenbau	1
2. Lieblingsgeschenk	3
3. Schlüsselanhänger	5
4. Baum fällt!	9
5. Schildkrötenpfad	11
6. Wasser auf die Mühle	15
7. Sack mit Münzen	19
8. Treffen sie sich?	23
9. Dottis	27
10. Erdbeerklau	31
11. Tiere beobachten	35
12. Geburtstagsrätsel	39
A. Aufgabenautoren	42
B. Sponsoring: Wettbewerb 2021	43
C. Weiterführende Angebote	46



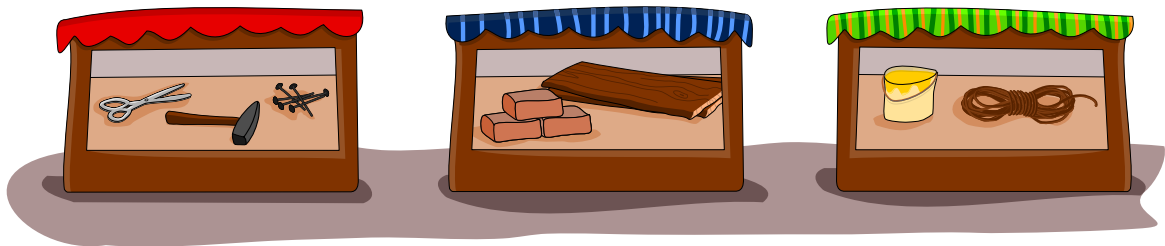
1. Brückenbau

Bella möchte eine Brücke bauen, über einen Bach. Sie braucht: Hammer, Nägel, Bretter und ein Seil. Im Keller findet sie einen Hammer und ein Seil.



Die anderen Sachen muss sie einkaufen. Unten siehst du drei Geschäfte, und was sie verkaufen.

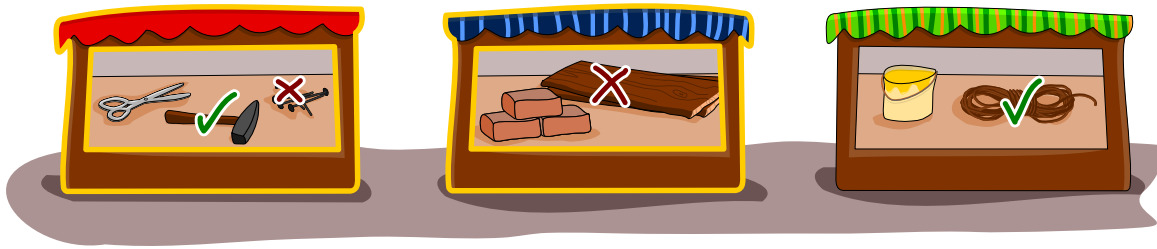
Wo kann Bella die anderen Sachen einkaufen?





Lösung

So ist es richtig:



Dies ist Informatik!

Die Geschäfte in dieser Biberaufgabe verkaufen insgesamt sieben Sachen: Schere, Hammer, Nägel, Ziegel, Bretter, Seil und Eimer. Das ist eine ganze Menge! Die beiden Sachen, die Bella einkaufen muss, sind ein *Teil dieser Menge*. Das kann man so malen: Man zeigt alle Sachen der gesamten Menge und markiert für jede Sache, ob sie zu Bellas Einkaufs-Teilmenge gehört oder nicht:



Genauso kann man malen, was die Geschäfte verkaufen, zum Beispiel das Geschäft links:



So sieht man auf den ersten Blick, was Bella im Geschäft links einkaufen kann: Die Nägel haben einen grünen Haken in der Einkaufs-Teilmenge und in der Verkaufs-Teilmenge.

Auch Computerprogramme müssen häufig Mengen vergleichen. Das können sie so machen wie oben gezeigt: Für jede Sache, die vorkommen kann, wird ein *Bit* benötigt. In einem Bit kann ein Computer einen von zwei Werten speichern, wie «ja» und «nein». In diesem Fall wird gespeichert, ob die Sache zu einer Menge gehört («ja») oder nicht («nein»). Dann kann ein Programm zwei Mengen so vergleichen: Es prüft, ob das Bit für eine Sache in der einen Menge «ja» ist und das Bit für die gleiche Sache in der anderen Menge auch «ja» ist. Eine solche Prüfung zweier Bits kann ein Computer besonders schnell erledigen. In der Informatik ist die Beschreibung von Mengen mit Bits deshalb sehr verbreitet.

Stichwörter und Webseiten

- Menge: [https://de.wikipedia.org/wiki/Menge_\(Datenstruktur\)](https://de.wikipedia.org/wiki/Menge_(Datenstruktur))
- Bits: <https://de.wikipedia.org/wiki/Bit>
- Bitstrings: <https://de.wikipedia.org/wiki/Bitkette>

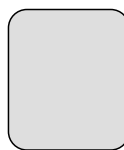


2. Lieblingsgeschenk

Die Biberfamilie hat drei Geschenke für ihre drei 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.



1: , 2: 



1: , 2: 

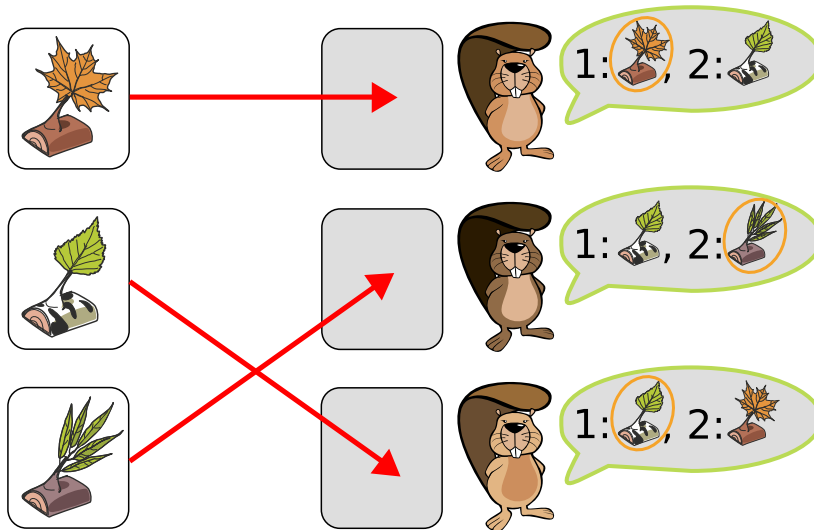


1: , 2: 



Lösung

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



Das dritte Geschenk wünscht sich nur der zweite Biber, deshalb muss er es bekommen. Ansonsten würde ein anderer etwas bekommen, das weder das Lieblingsgeschenk noch das zweitliebste ist. Für die anderen beiden Geschenke ist die Aufteilung dann klar: Jeder kann sein Lieblingsgeschenk bekommen.

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 Problem 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? Informatik ist eben ein ganz vielseitiges Fach.

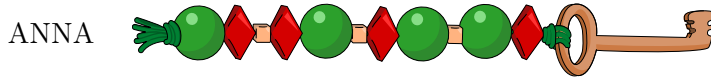
Stichwörter und Webseiten

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

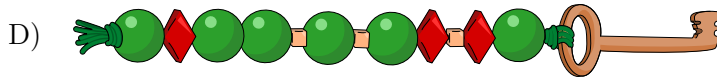
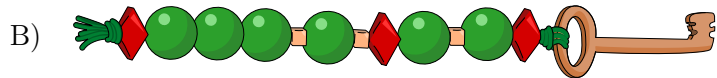
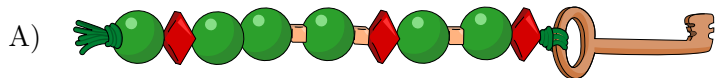


3. Schlüsselanhänger

ANNA, BELLA und LENA machen Schlüsselanhänger mit ihren Namen. Für die Buchstaben verwenden sie zwei Sorten von Perlen: ● und ◆. Einzelne Buchstaben trennen sie mit dieser Perle: ■.



Welchen Schlüsselanhänger hat LENA gemacht?





Lösung

Die richtige Antwort ist A)

Das Wort LENA beginnt mit L. In BELLA ist L der dritte und vierte Buchstabe und wir erkennen in BELLA für L die Perlenfolge . Nur Antworten A) und D) beginnen mit diesem Buchstaben und können die Lösung sein. Der zweite Buchstabe von LENA, also E, ist auch der zweite von BELLA, dort finden wir im Anhänger jeweils die Perle . Sowohl bei A) als auch bei D) haben wir als zweiten Buchstaben, also könnten sie beide weiterhin die Lösung sein. Als nächstes wollen wir die Perlen für den Buchstaben N herausfinden. In ANNAs Anhänger finden wir für N die Perlenfolge . Und das sind nur in der Lösung A) die nächsten Perlen.

Eine andere Möglichkeit, die Perlen für LENAs Anhänger herauszufinden, ist es, eine Tabelle mit den Perlen für die bereits bekannten Buchstaben zu erstellen. Von ANNAs Anhänger finden wir für A die Perlenfolge und für N . Von BELLAs Anhänger finden wir die Perlenfolge für B: , für E: und L: .

Buchstaben	Perlen
A	
N	
B	
E	
L	

Daher können wir den Anhänger für LENA aus den Perlen , und formen, wenn wir zusätzlich die einzelnen Buchstaben mit der Perle trennen. So erhalten wir den Anhänger , was die Antwort A) ist. Wenn wir die anderen gegebenen Anhänger mit Hilfe der Codierungstabelle entschlüsseln, finden wir für B) BENA, C) NENA und für D) LEAN.

Dies ist Informatik!

Information wird *codiert*, um Nachrichten unter bestimmten Bedingungen übertragen zu können oder um die Information geheim (*verschlüsselt*) zu übertragen. In dieser Aufgabe basiert die Codierung auf dem Morsecode. Der Punkt • des Morsecode wird durch die runde Perle dargestellt und der Strich – durch . Der Buchstabe A ist im Morsecode •–, also im Perlencode . Um beliebige Texte zu codieren, brauchen wir Codes für alle Buchstaben des Alphabets. Der Morsecode entstand im 19. Jahrhundert. Samuel Morse erfand 1837 einen einfachen elektromagnetischen Schreibtelegraphen. Der verwendete Code umfasste damals nur die zehn Ziffern 0 bis 9; die übertragenen Zahlen mussten mit Hilfe einer Tabelle in Buchstaben und Wörter übersetzt werden. Alfred Lewis Vail, ein Mitarbeiter von Morse, entwickelte ab 1838 den ersten Code, der auch Buchstaben umfasste. Der Code wurde entwickelt, um Texte akustisch, optisch oder elektrisch über Telegraphenleitungen zu übertragen. Ein Punkt ist eine kurze Übertragungsdauer und ein Strich eine dreimal so lange. Zwischen den Buchstaben muss eine Pause eingehalten werden. Eine längere Pause trennt Wörter. Noch heute wird



der Morsecode für das Notsignal SOS verwendet. SOS im Morsecode **••• — — — •••** (3x kurz, 3x lang, 3x kurz) kann sehr leicht durch Rufen, Klopfen, oder mit der Taschenlampe übertragen werden. In der elektronischen Datenverarbeitung werden Zeichen über einen Zahlenwert codiert, um sie zu übertragen oder zu speichern.

Stichwörter und Webseiten

- Zeichenkodierung: <https://de.wikipedia.org/wiki/Zeichenkodierung>
- Morsecode: <https://de.wikipedia.org/wiki/Morsecode>





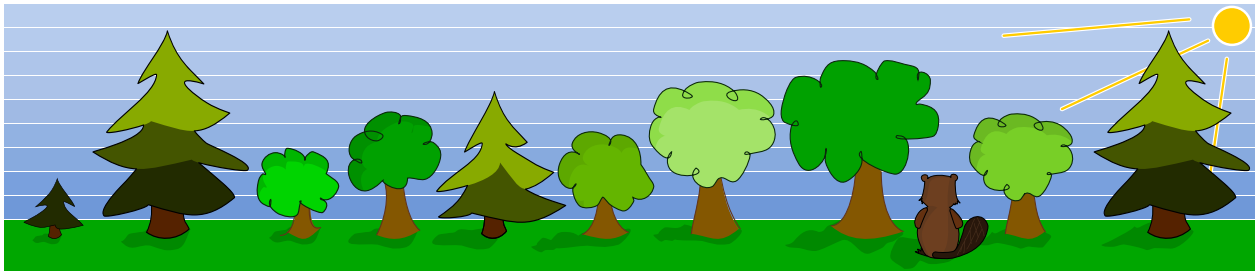
4. Baum fällt!

Ein Biber möchte einen Damm bauen. Damit er immer die richtigen Bäume fällt, hat er sich zwei Bedingungen überlegt:

Er fällt einen Baum nur dann, wenn ...

- ... direkt links daneben ein kleinerer Baum wächst und ...
- ... direkt rechts daneben ein grösserer Baum wächst.

Welche Bäume wird der Biber fällen?





Lösung

Nur die Bäume an vierter Stelle und siebter Stelle erfüllen die vorgegebenen Bedingungen: Direkt links befindet sich ein kleiner Baum UND direkt rechts ein grösserer Baum.



Dies ist Informatik!

In der Informatik geht es oft darum, Probleme zu lösen, die durch eine Reihe von logischen *Einschränkungen* (engl. *constraints*) spezifiziert sind. Die Aufgabe besteht darin, eine Lösung zu finden, die alle gegebenen Einschränkungen erfüllt. Komplexere Aufgaben als diese können behandelt werden, indem die Einschränkungen mit Hilfe von *logischen Operatoren* kombiniert werden. Die Konjunktion (\wedge oder auch UND-Operator) beispielsweise liefert im Ausdruck $A \wedge B$ genau dann «wahr» als Ergebnis, wenn die beiden Einschränkungen ebenfalls wahr sind. In dieser Aufgabe wäre das dann also: «Der Baum links ist kleiner.» \wedge «Der Baum rechts ist grösser.» Dieses grundlegende Prinzip kommt in nahezu allen Bereichen der Informatik vor, wie beispielsweise bei vielen Sortier-Algorithmien, zum Beispiel dem *Bubble Sort*, hier werden immer mehrere Objekte einer Liste auf gewisse Einschränkungen überprüft, um sie danach, sofern notwendig, an eine andere Position in der Liste zu verschieben. Dieses Prinzip wird solange wiederholt, bis die Liste vollständig sortiert ist.

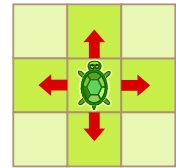
Stichwörter und Webseiten

- Algorithmisches Denken (algorithmic thinking)
- logische Operatoren: https://de.wikipedia.org/wiki/Logischer_Operator
- Sortieren: <https://sorting.at/>
- Constraint Satisfaction Problem:
<https://de.wikipedia.org/wiki/Constraint-Satisfaction-Problem>



5. Schildkrötenpfad

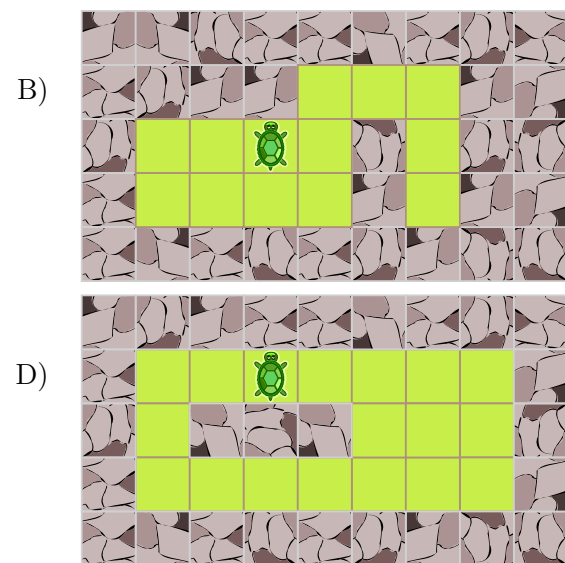
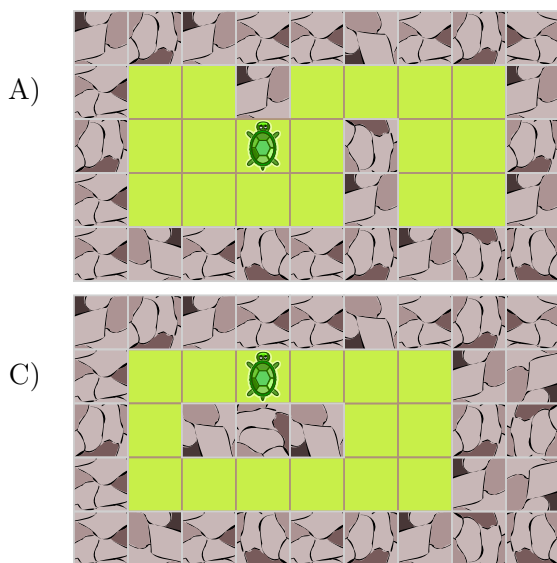
Eine Schildkröte soll verschiedene Gärten abgrasen. Jeder Garten ist in Quadrate unterteilt, die entweder mit Gras oder Steinen bedeckt sind. Die Schildkröte kann keine Steine überqueren. Sie kann sich aber von einem Grasfeld zu einem anderen Grasfeld direkt daneben bewegen.

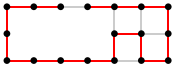


Die Schildkröte soll die Gärten vollständig abgrasen. Sie startet auf dem Feld, auf dem sie im Bild steht. Am Ende soll sie in jedem Grasfeld genau einmal gewesen sein.

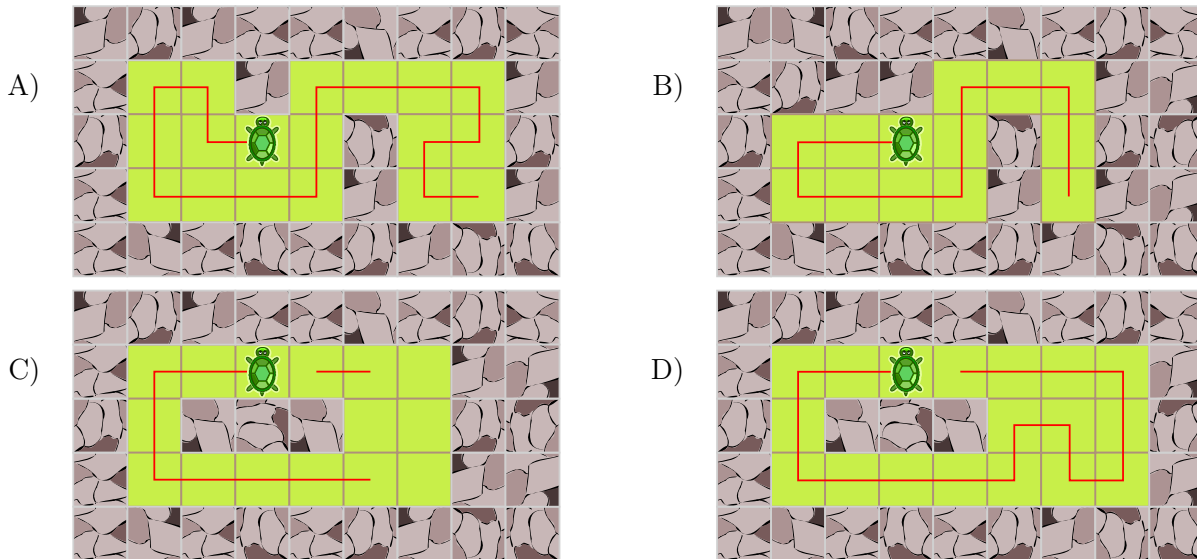
Leider kann die Schildkröte so einen der Gärten nicht vollständig abgrasen.

Welcher ist es?





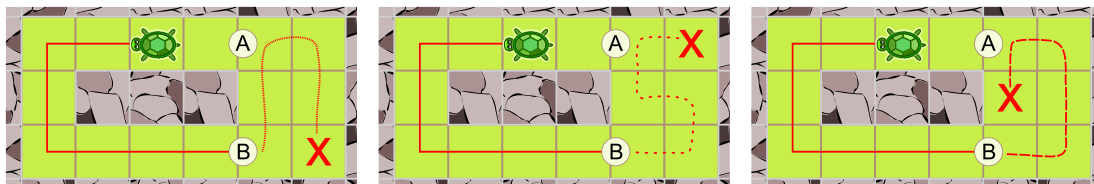
Lösung



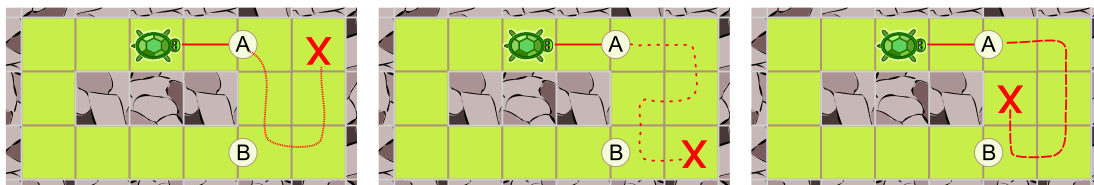
Die Gärten A, B und D kann die Schildkröte vollständig abgrasen.

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.

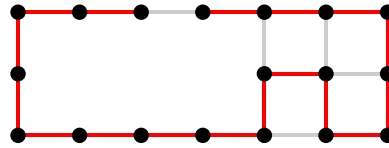


Dies ist Informatik!

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>



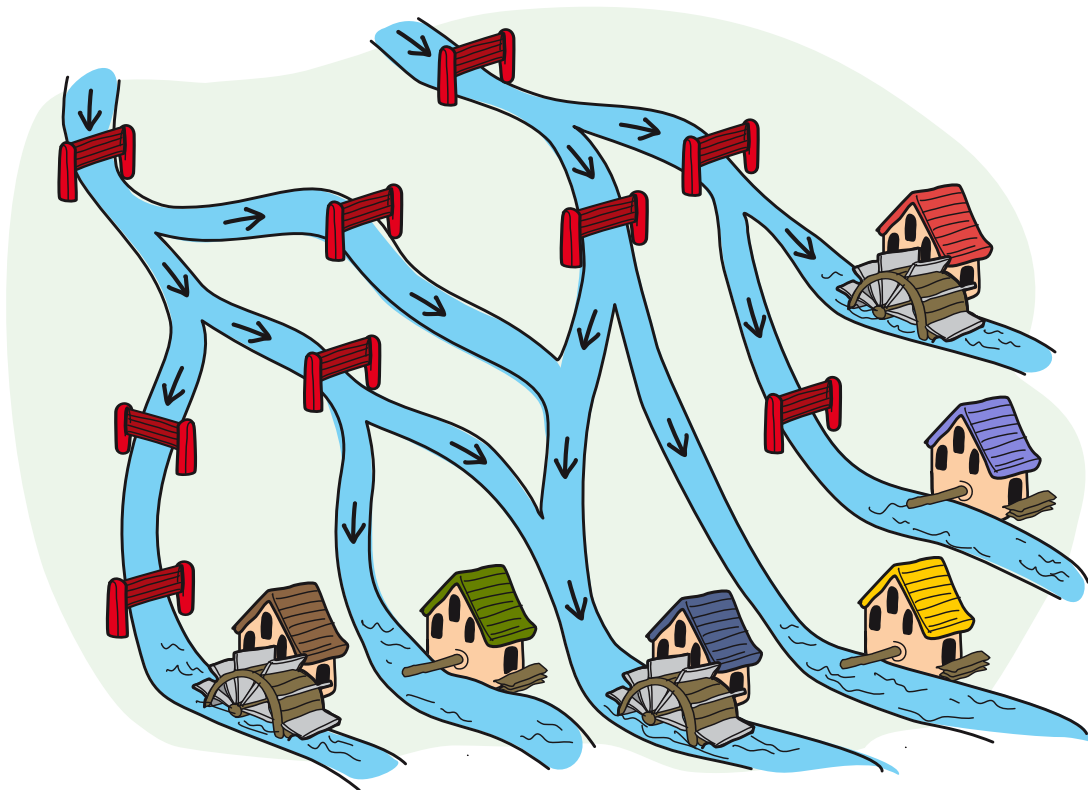


6. Wasser auf die Mühle

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.

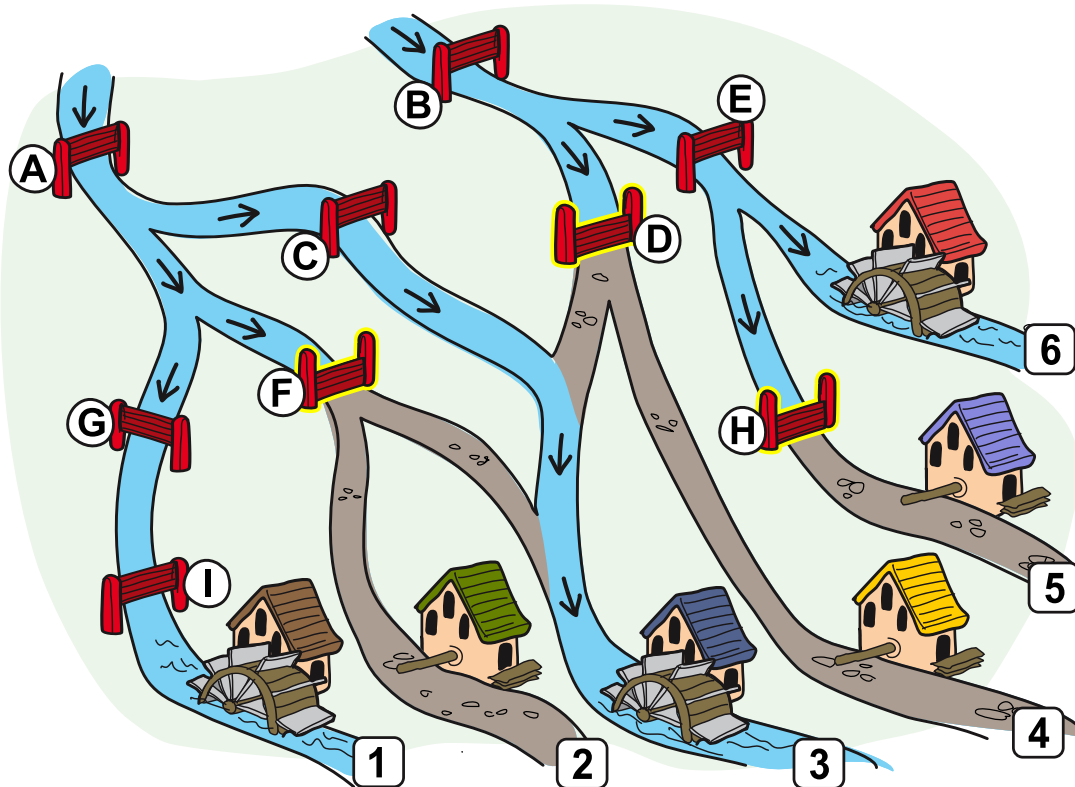
Welche Schieber soll Mert schliessen?





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

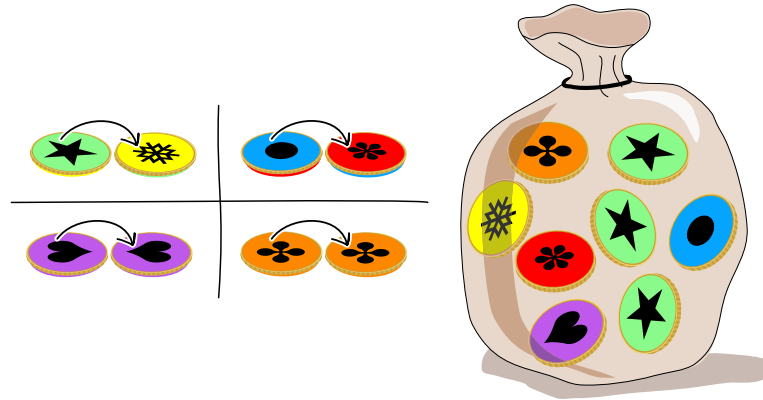
- Bedingte Anweisung: https://de.wikipedia.org/wiki/Bedingte_Anweisung_und_Verzweigung
- Boolsche Variable: <https://de.wikipedia.org/wiki/Boolean>
- Boolesche Operatoren: https://de.wikipedia.org/wiki/Boolescher_Operator





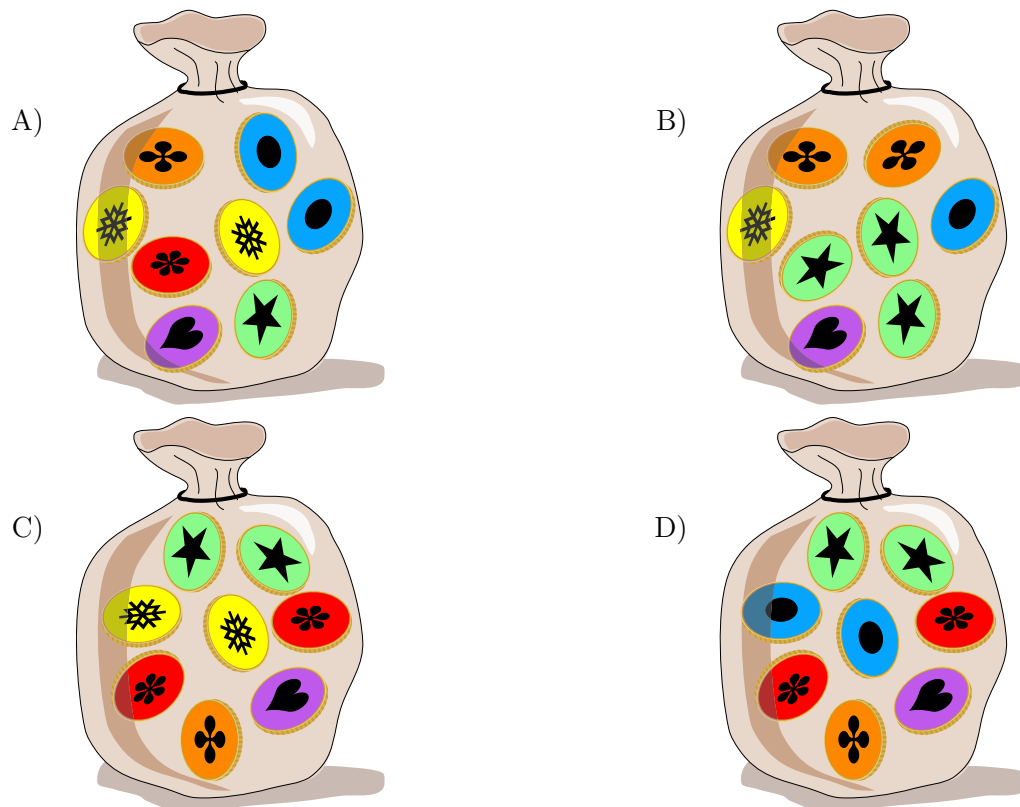
7. Sack mit Münzen

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.

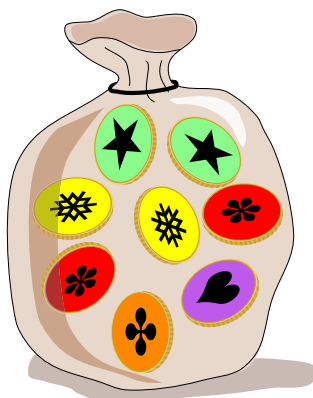
Welcher ist Emils Sack?



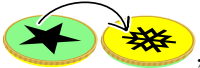





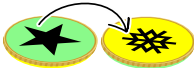



Lösung

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

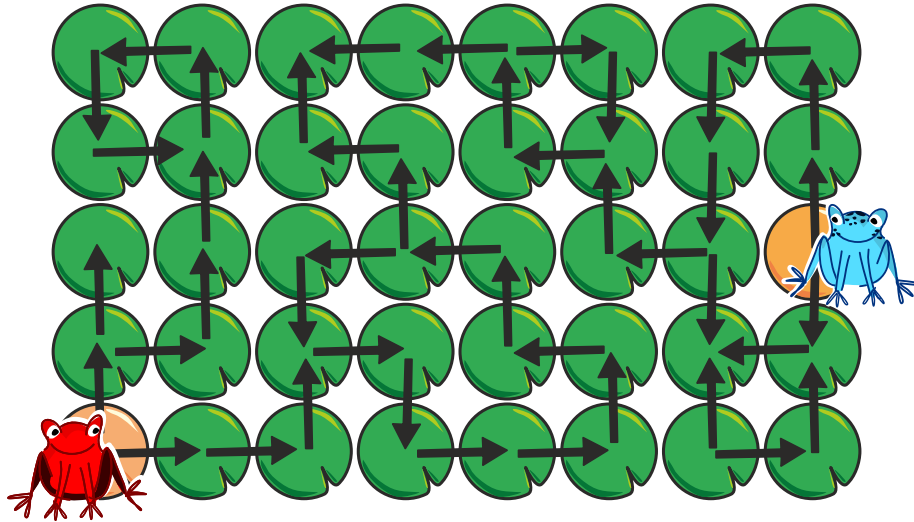




8. Treffen sie sich?

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

Auf welchem Seerosenblatt können sie sich treffen?

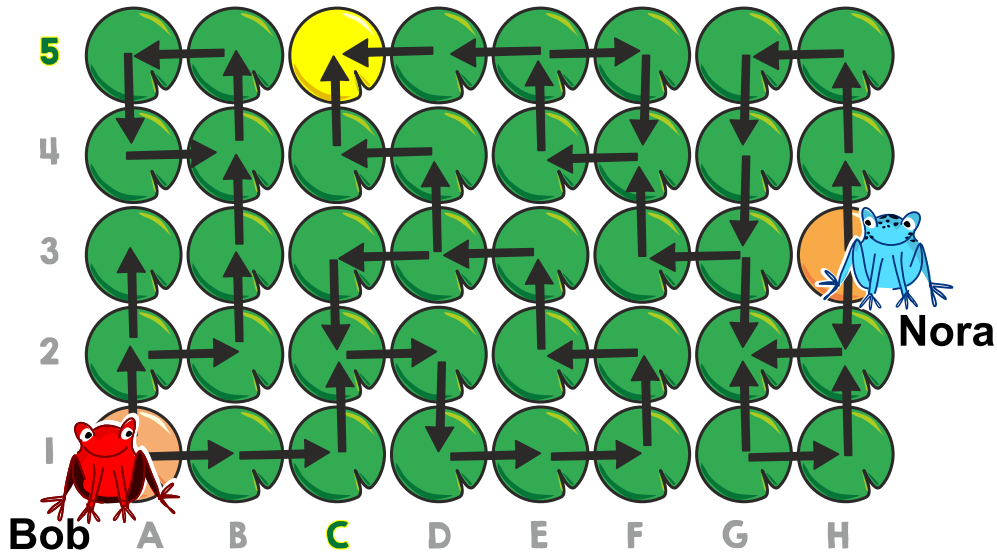


Man kann auf die Blätter klicken. Klickt man auf ein Blatt, wird dieses ausgewählt und gleichzeitig ein bereits ausgewähltes Blatt wieder deaktiviert.



Lösung

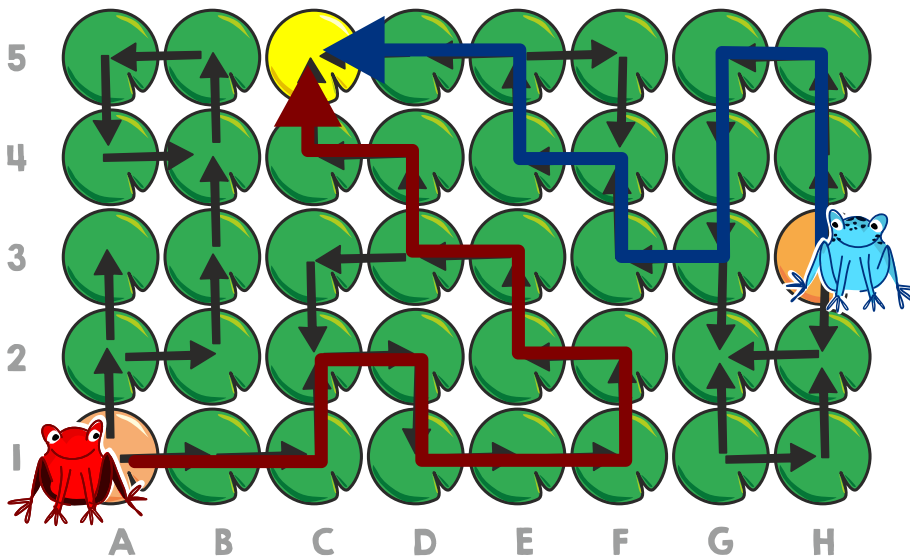
Die Frösche können sich nur auf dem Blatt C5 treffen.



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.

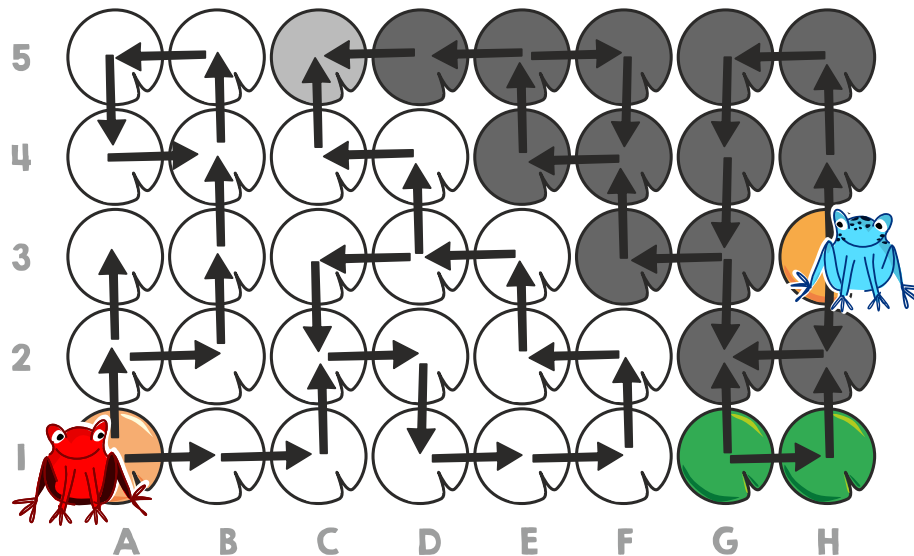
Der blaue Frosch Nora hat am Start auch zwei Wahlmöglichkeiten. Geht sie nach «unten», gerät sie in die Sackgasse G2. Falls sie nach «oben» startet, erreicht sie zunächst G3. Von dort kann sie entweder wieder in die Sackgasse G2 geraten oder nach «links» gehen und schliesslich E5 erreichen. Von dort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





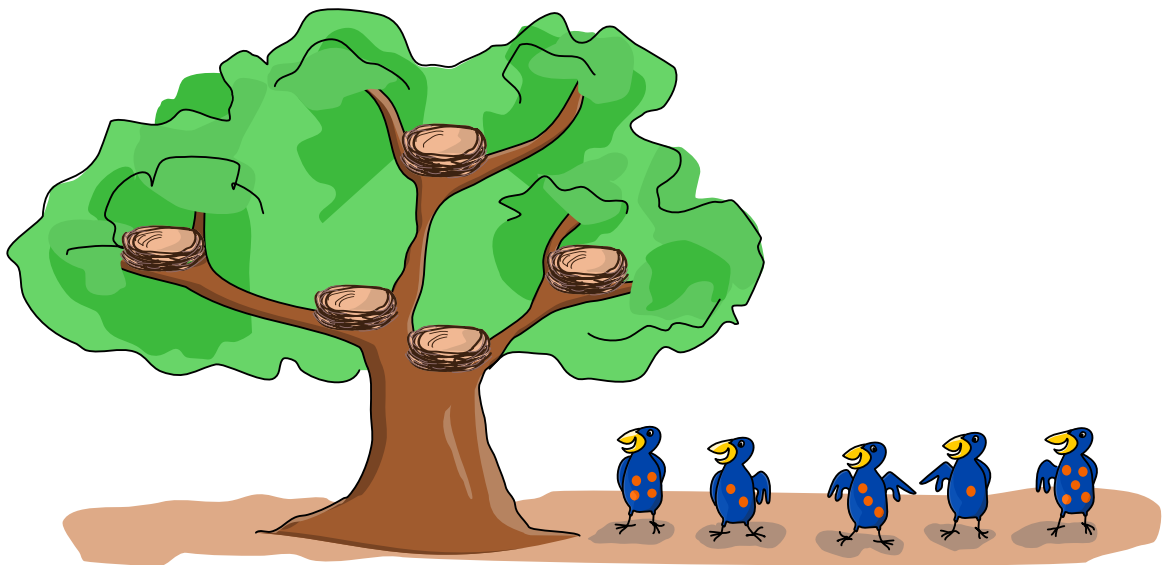
9. Dottis

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.

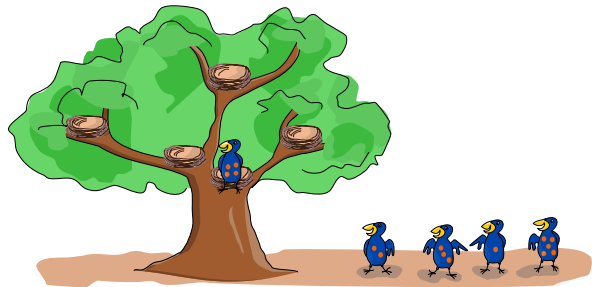




Lösung

So kommt man auf die richtige Lösung:

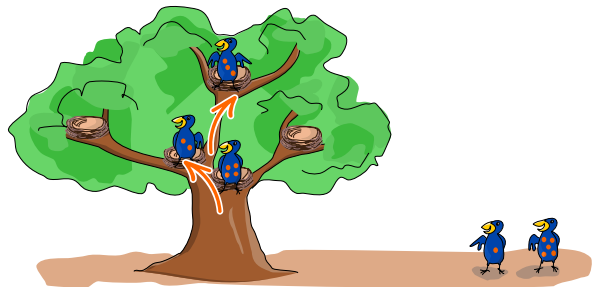
Der erste Dotti, der mit 4 Punkten, zieht in das unterste Nest und bleibt dort.



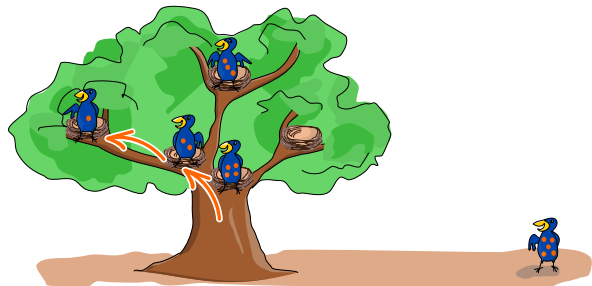
Der zweite Dotti hat 2 Punkte. Im untersten Nest sitzt der erste Dotti mit 4 Punkten. Weil 4 mehr als 2 ist, klettert der zweite Dotti nach links weiter und zieht in das erste freie Nest.



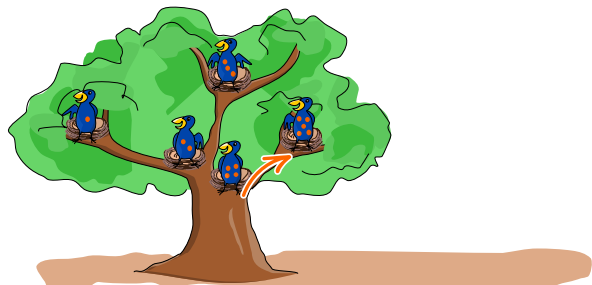
Der dritte Dotti hat 3 Punkte. Dieser klettert beim untersten Nest, wo der Dotti mit 4 Punkten sitzt, nach links, weil 4 mehr als 3 ist. Im nächsten Nest sitzt der Dotti mit 2 Punkten. Weil 3 mehr als 2 ist, klettert der dritte Dotti nach rechts weiter. Er zieht dann in das nächste freie Nest. Das ist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

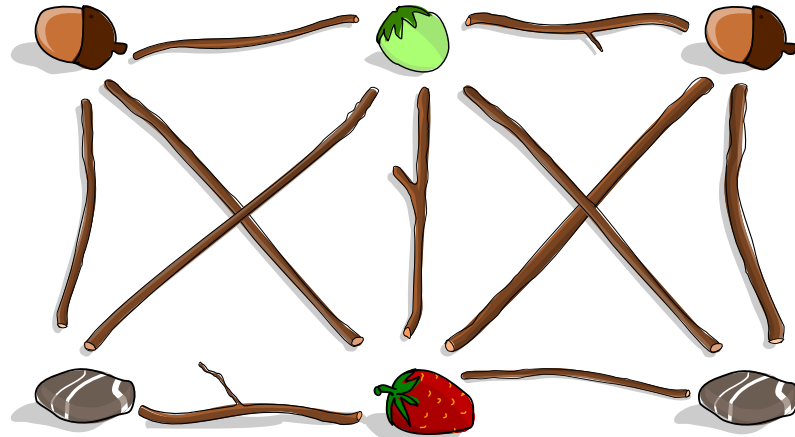




10. Erdbeerklau

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.

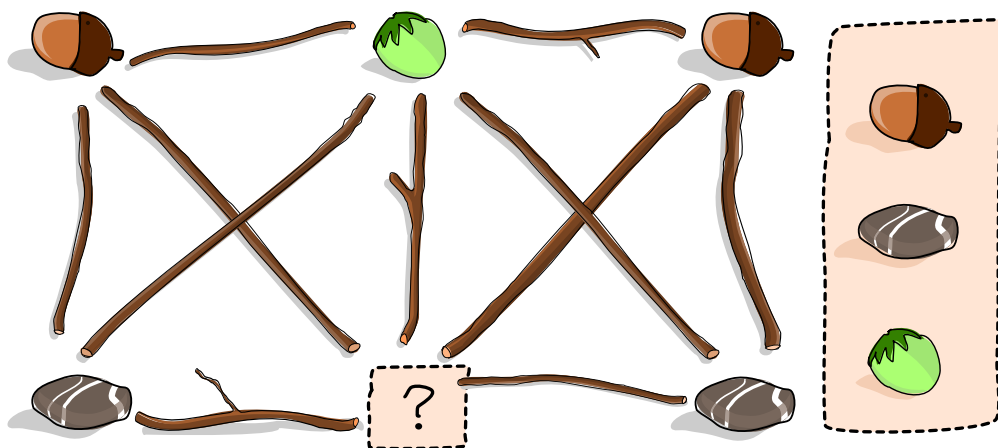
Danach legt Anja Äste zwischen diese Sachen. Dabei befolgt sie folgende Regel: Ein Ast darf nicht zwischen zwei gleichen Sachen liegen – zum Beispiel nicht zwischen zwei Eicheln. Hier ist das fertige Kunstwerk:



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

Kannst du ihm helfen, die Tat zu verschleiern?

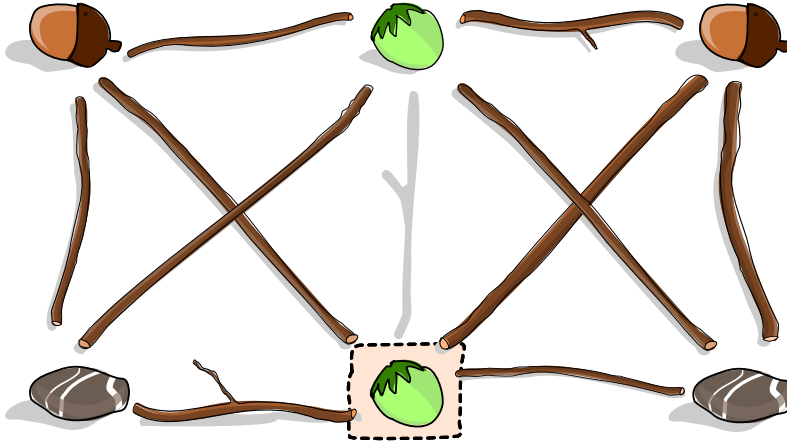
Platziere eine andere Sache an die Stelle der Erdbeere und entferne genau einen Ast. Am Ende soll Anjas Regel auch für das veränderte Kunstwerk gelten.





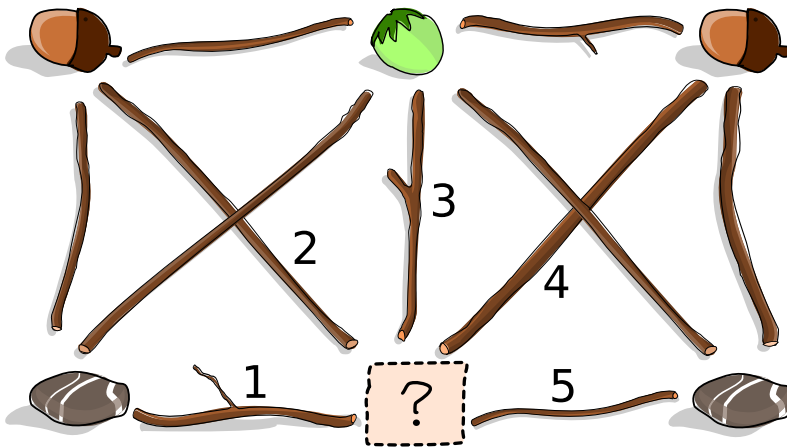
Lösung

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 Cliques mit vier Knoten: die rechte und die linke Hälfte des Graphen. (Die Haselnuss oben und das Fragezeichen gehören zu beiden Cliques.) 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 Cliques mit höchstens 3 Knoten übrig bleiben, wenn die Regel weiterhin erfüllt bleiben soll. Es muss also eine Kante (ein Ast) entfernt werden, sodass beide Cliques 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 benutzen. 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

- Färbepblem: [https://de.wikipedia.org/wiki/Färbung_\(Graphentheorie\)](https://de.wikipedia.org/wiki/Färbung_(Graphentheorie))
- minimale Kantenfärbung, Kantenfärbung: <https://de.wikipedia.org/wiki/Kantenfärbung>
- Clique: [https://de.wikipedia.org/wiki/Clique_\(Graphentheorie\)](https://de.wikipedia.org/wiki/Clique_(Graphentheorie))

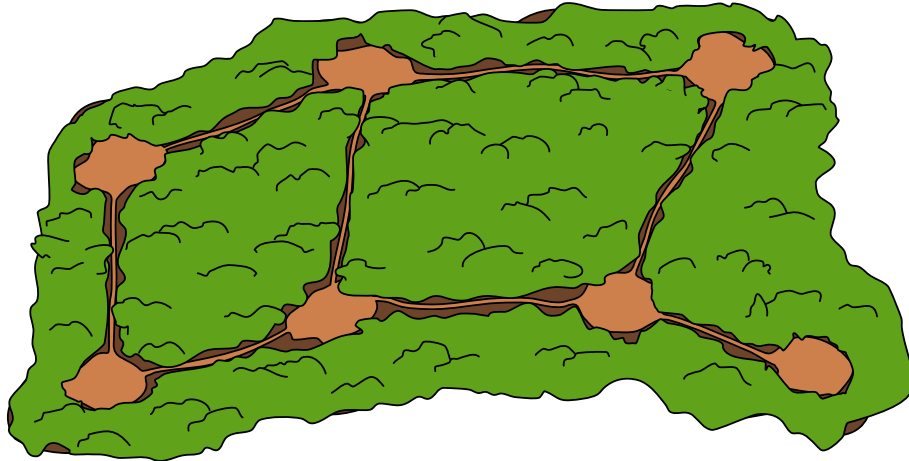




11. Tiere beobachten

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.

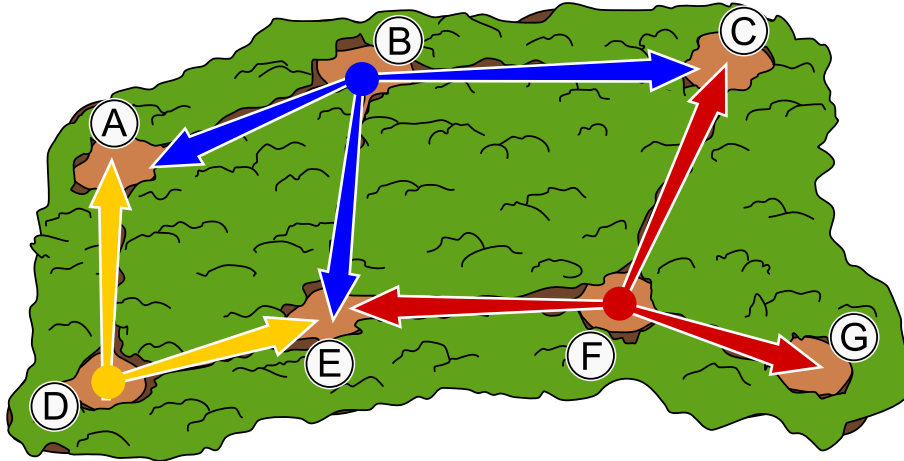
Wähle möglichst wenige Lichtungen, von denen die Förster alle Wege beobachten können!





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.

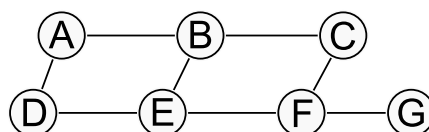
Es sind also mindestens drei Förster zur Beobachtung aller Wege notwendig. Daher ist die hier angegebene Lösung eine Lösung mit der kleinstmöglichen Anzahl an Förstern. Tatsächlich gibt es keine weitere Lösung mit genau drei Förstern.

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

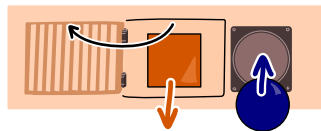
- Graph: [https://de.wikipedia.org/wiki/Graph_\(Graphentheorie\)](https://de.wikipedia.org/wiki/Graph_(Graphentheorie))
- Knotenüberdeckung: <https://de.wikipedia.org/wiki/Knotenüberdeckung/>





12. Geburtstagsrätsel

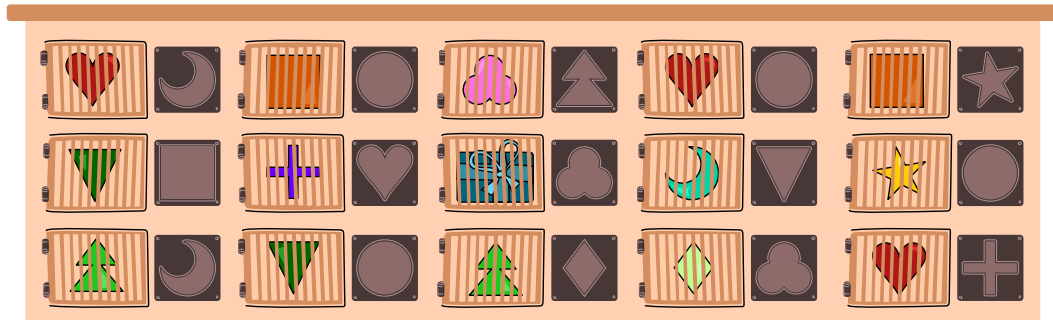
Bastian bekommt zum Geburtstag eine Kiste mit 15 Türen. Hinter der mittleren Tür ist ein weiteres Geschenk. Hinter den anderen Türen sind Bausteine. Zu jeder Tür gehört ein Loch, rechts neben der Tür. Bastian kann eine Tür öffnen, indem er in das Loch einen Baustein gleicher Form einwirft – wie einen Schlüssel.



Zu Beginn hat Bastian diesen runden Baustein: 

Er will höchstens fünf Türen öffnen, um das Geschenk zu erreichen.

Welche Tür muss Bastian dafür zuerst öffnen?



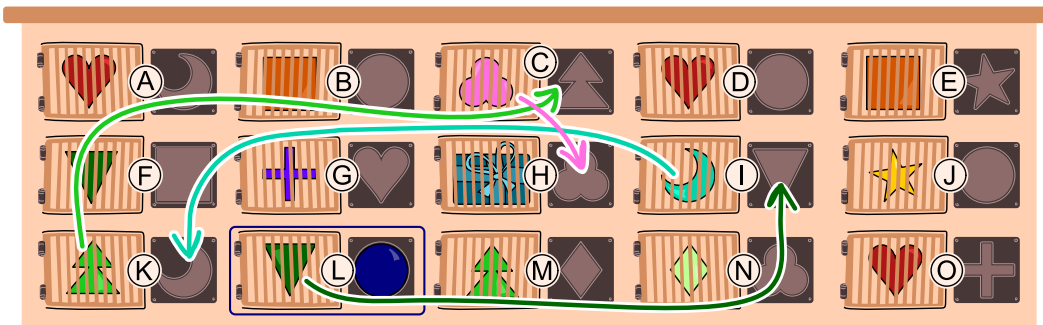


Lösung

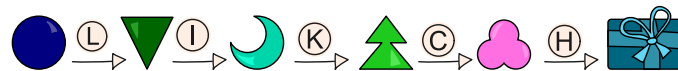
Bastian muss zuerst die blau markierte Tür öffnen:



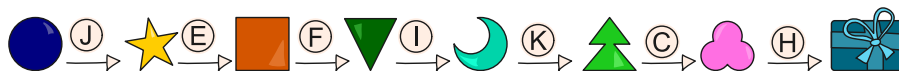
Im folgenden Bild sind die Türen mit Buchstaben versehen und die Pfeile zeigen, wie Bastian mit insgesamt 5 Türöffnungen das Geschenk erreicht.



Wir können die Reihenfolge, in der er die fünf Türen öffnet, auch wie folgt darstellen.



Es gäbe auch andere Wege zum Geschenk, zum Beispiel folgenden.

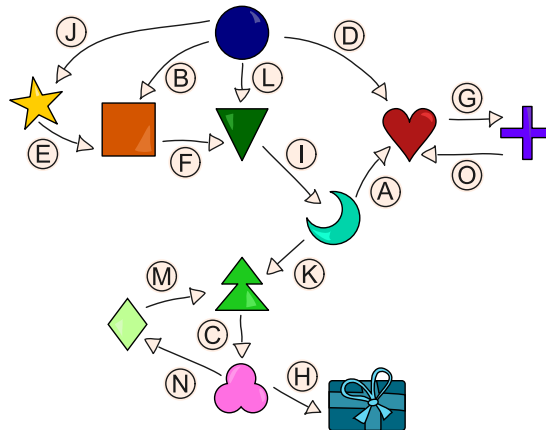


Doch diese Wege sind alle zu lang, es müssten mehr als fünf Türen geöffnet werden. Alle Möglichkeiten durchzuprobieren, ist ziemlich aufwendig.

Im vorliegenden Fall findet man den kürzesten Weg und damit die richtige Lösung am schnellsten mit einer sogenannten *Rückwärtssuche*: Man beginnt bei der Tür mit dem Geschenk und schaut dann jeweils, welchen Baustein man benötigt.

Dies ist Informatik!



Mit mehr Zeit und Aufwand können wir die Situation in der Aufgabe auch als *Graph* darstellen:



Ein Graph besteht allgemein aus *Knoten* (Kreisen) und *Kanten* (Linien) zwischen den Knoten. In unserem Fall haben wir einen Knoten für jede Form und das Geschenk. Die Kanten sind hier Pfeile (auch *gerichtete* Kanten genannt) und entsprechen den Türen. Jeder Pfeil führt von der Form zum Öffnen der Tür zu der Form hinter der Tür.

Die Informatik arbeitet sehr gerne mit Graphen. Einerseits bieten sie oft anschauliche Darstellungen von abstrakten Zusammenhängen.

Andererseits existieren fertige Algorithmen, die uns Fragen zu Graphen sehr effizient beantworten. Bei komplizierteren Aufgaben kann sich der Aufwand für das Aufstellen des Graphen deshalb schnell lohnen.


In der vorliegenden Aufgabe suchen wir einen Weg der Länge höchstens 5 vom erhaltenen Baustein  zum Geschenk . Ein guter Algorithmus dafür ist die sogenannte *Breitensuche*. Dieser funktioniert sowohl für Graphen mit gerichteten Kanten, wie in der Aufgabe, also auch für Graphen mit ungerichteten Kanten.

Stichwörter und Webseiten

- Gerichteter Graph: https://de.wikipedia.org/wiki/Gerichteter_Graph
- Breitensuche: <https://de.wikipedia.org/wiki/Breitensuche>



A. Aufgabenautoren


 Sarah Atkins


 Michael Barot

 Liam Baumann


 Linda Björk Bergsveinsdóttir


 Sarah Chan

 Valentina Dagiéné

 Christian Datzko


 Susanne Datzko

 Nora A. Escherle

 Margarita Flores-Sicich

 Fabian Frei

 Gerald Futschek


 Yasemin Gülbahar

 Ezgi Arzu Güneş

 Benjamin Hirsch

 Juraj Hromkovič

 Andrea Hrušecká

 Tiberiu Iorgulescu

 Ungyeol Jung


 Filiz Kalelioğlu

 Martin Kandlhofer

 Vaidotas Kinčius

 V́ictor Koleszar

 Regula Lacher

 Taina Lehtimäki

 Marielle Léonard

 Tom Naughton


 Graciela Oyhenard

 Jean-Philippe Pellet

 Zsuzsa Pluhár

 Wolfgang Pohl

 Peter Rossmann

 Eljakim Schrijvers

 Rosario Schunk

 Troy Vasiga

 Florentina Voboril

 Michael Weigend



B. Sponsoring: Wettbewerb 2021

HASLERSTIFTUNG

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

Stiftungszweck der Hasler Stiftung ist die Förderung der Informations- und Kommunikationstechnologie (IKT) zum Wohl und Nutzen des Denk- und Arbeitsplatzes 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



OXOCARD

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

educaTEC

<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.

**senarclens
leu+partner**
strategische kommunikation

<http://senarclens.com/>
Senarclens Leu & Partner

ABZ
AUSBILDUNGS- UND BERATUNGSZENTRUM
FÜR INFORMATIKUNTERRICHT

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

hep/ haute
école
pédagogique
vaud

<http://www.hep1.ch/>
Haute école pédagogique du canton de Vaud

PH LUZERN
PÄDAGOGISCHE
HOCHSCHULE

<http://www.phlu.ch/>
Pädagogische Hochschule Luzern

n|w Fachhochschule
Nordwestschweiz

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

Scuola universitaria professionale
della Svizzera italiana

SUPSI

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



PÄDAGOGISCHE
HOCHSCHULE
ZÜRICH



<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.

010100110101011001001001
010000010010110101010011
010100110100100101000101
001011010101001101010011
010010010100100100100001



www.svia-ssie-ssii.ch
schweizerischervereinfürinformatikind
erausbildung//sociétésuissepourl'infor
matique dans l'enseignement//sociétésviz
zeraperl'informatice nell'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.