INFORMATIK-BIBER SCHWEIZ
CASTOR INFORMATIQUE SUISSE
CASTORO INFORMATICO SVIZZERA

STIU

Informatik-Biber unplugged für 3. – 6. Klasse

Land	ID	**Titel**	3/4	5/6	7/8	9/10	11–13
	2017-CA-01	**Parkplatz**	leicht	leicht	–	–	
	2021-SK-01	**Schlüsselanhänger**	mittel	leicht	–	–	–
	2016-DE-09	**Passende Paare**	mittel	leicht	–	–	–
	2019-CH-03b	**Beavercoins**	mittel	leicht	–	–	–
	2018-HU-01	**Adas Farbstifte**	mittel	–	–	–	–
	2017-SK-04	**Fünf Hölzli**	schwer	mittel	–	–	–
	2016-CH-05a	**Binärer Geburtstag**	–	mittel	leicht	–	–
	2019-IN-09	**Rangoli**	–	mittel	leicht	leicht	–
	2019-SK-04	**Zeichenroboter**	schwer	mittel	leicht	–	–
	2019-IN-09	**Rangoli**	–	mittel	leicht	leicht	–
	2016-CH-12	**Blumen und Sonnen**	–	schwer	leicht	–	–
	2017-AT-06	**Klammeranhänger**	–	schwer	leicht	–	–
	2019-CH-10	**Nachricht der Urbiber**	–	schwer	mittel	–	–
	2019-RU-01	**Abwaschmaschine einräumen**	–	–	leicht	–	–
	2019-CH-10	**Nachricht der Urbiber**	–	schwer	mittel	–	–
	2021-CH-04c	**Erdbeerklau**	–	schwer	mittel	leicht	–
	2021-CA-01b	**Dotties**	–	schwer	mittel	leicht	–
	2017-JP-02	**Japanischer Stockkampf**	–	schwer	mittel	leicht	–
	2016-FR-03	**Brückenbau**	–	–	mittel	leicht	–
	2019-RU-01	**Abwaschmaschine ausräumen**	–	–	leicht	–	–
	2015-SE-01	**Stellas Sterne**	–	schwer	mittel	leicht	–
	2016-IS-02	**Gruppenarbeit**	–	schwer	mittel	leicht	–
	2016-CH-23	**Nim**	–	leicht	mittel	schwer	schwer
	2017-CH-07b	**Gänge im Biberbau**	–	leicht	mittel	schwer	schwer
	2016-FR-02	**Verkehrshüetli-Versteck**	–	–	–	schwer	–
	2019-HU-02	**Ein Sack voller Bonbons**	–	–	schwer	mittel	leicht
	2018-BE-01a	**Karten drehen**	–	–	–	–	mittel
	2015-SI-07	**Piratenjagd**	–	–	–	schwer	schwer

INFORMATIK-BIBER SCHWEIZ
CASTOR INFORMATIQUE SUISSE
CASTORO INFORMATICO SVIZZERA

Kontakt: biber@informatik-biber.ch, christian@datzko.ch, susanne@datzko.ch.

Die folgenden Autoren haben an den Aufgaben mitgewirkt:

Andrea Adamoli
Tony René Andersen
Guðjón Karl Arnarson
Michelle Barnett
Michael Barot
Wilfried Baumann
Khuyagbaatar Batsuren
Daphne Blokhuis
Nicolas Brunner
Špela Cerar
Sarah Chan
Marios Choudary
Anton Chukhnov
Kris Coolsaet
Allira Crowe
Valentina Dagienė
Christian Datzko
Sarah Estrella Datzko
Susanne Datzko
Janez Demšar
Marissa Engels
Olivier Ens
Hanspeter Erni
Nora Anna Escherle
Gerald Futschek
Fabian Frei
Sonali Gogate
Arnheiður Guðmundsdóttir
Martin Guggisberg
Urs Hauser
Fredrik Heintz
Mathias Hiron
Juraj Hromkovič
Andrea Hrušecká
Thomas Ioannou
Anna Laura John
Ungyeol Jung
Tobias Kohn
Ivana Kosirova
Bernd Kurzmann
Vaidotas Kinčius
Iryna Kirynovich
Mária Kiss
Regula Lacher
Inggriani Liem
Violetta Lonati
Hiroki Manabe
Mattia Monga
Anna Morpurgo
Madhavan Mukund
Tom Naughton
Pia Niemelä
Tomohiro Nishida
Henry Ong
Zsuzsa Pluhár
Wolfgang Pohl
Sergey Pozdnyakov
Stavroula Prantsoudi
Nol Premasathian
J.P. Pretti
Milan Rajković
Alei Reyes
Chris Roffey
Fran Rosamond
Kirsten Schlüter
Eljakim Schrijvers
Vipul Shah
Maiko Shimabuku
Taras Shpot
Pär Söderhjelm
Björn Steffen
Monika Tomcsányiová
Peter Tomcsányi
Ahto Truu
Nicole Trachsler
Jiří Vaníček
Troy Vasiga
Rechilda Villame
Michael Weigend
Eslam Wageed
Momo Yokoyama
Khairul Anwar Mohamad Zaki

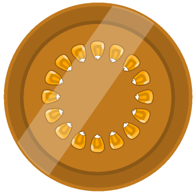
https://www.informatik-biber.ch/ https://www.castor-informatique.ch/ https://castoro-informatico.ch/

Beavercoins

2019-CH-03b

3/4	5/6	7/8	9/10	11–13
mittel	leicht	–	–	–

Im Biberland verwendet man „Beavercoins“ als Währung. Die Münzen haben die folgenden Werte:

				
16	8	4	2	1

Die Biber tragen nicht gerne viele Münzen bei sich und zahlen deswegen mit so wenig Münzen wie möglich.

Mit welchen Münzen würdest Du 13 Beavercoins bezahlen, wenn Du möglichst wenige Münzen verwendest?

Lösung

Die beste und damit richtige Lösung ist, mit ,  und  zu bezahlen, also mit einer 8-Beavercoins-Münze, einer 4-Beavercoins-Münze und einer 1-Beavercoins-Münze. Die Summe der Münzen ergibt $8 + 4 + 1 = 13$. Mit weniger Münzen ist es nicht möglich, denn eine Münze grösser als die 8-Beavercoins-Münze wäre bereits die 16-Beavercoins-Münze und es gibt keine Münze mit dem Wert der noch fehlenden 5 Beavercoins. Die nächstkleinere Münze ist die 4-Beavercoins-Münze, so dass man zusammen mit der 1-Beavercoins-Münze eben diese drei Münzen braucht.

Um die richtige Lösung zu finden, kann man auch mit einer anderen Kombination anfangen, beispielsweise mit zwei 4-Beavercoins-Münzen, einer 2-Beavercoins-Münze und drei 1-Beavercoins-Münzen. Als nächstes kann man solange zwei Münzen mit demselben Wert durch eine Münze mit doppeltem Wert ersetzen, bis man zum richtigen Ergebnis kommt.

Das ist Informatik!

Informatikerinnen und Informatiker sind Experten dafür, Informationen als Folge von Symbolen darzustellen. Dazu gehört auch das Darstellen von Zahlen. In dieser Aufgabe geht es darum, dass ein Geldbetrag mit verschiedenen Kombinationen von Münzen bezahlt werden kann. Diese Kombination ist nicht eindeutig, verschiedene Kombinationen mit Münzen unterschiedlicher Werte ergeben denselben Geldbetrag. Daher geht es in dieser Aufgabe auch darum, die eine Kombination mit der geringsten Anzahl von Münzen herauszufinden.

Die Münzen in dieser Aufgabe sind so gewählt, dass zwei Münzen von gleichem Wert zusammen immer dem Wert der nächstgrösseren Münze entsprechen. Das ergibt das *binäre Zahlensystem* mit den Stellenwerten 1, 2, 4, 8, 16 und so weiter. Im binären Zahlensystem ist die Darstellung einer beliebigen Zahl wie der 13 immer eindeutig: ein Stellenwert ist entweder verwendet oder nicht.

Ähnlich funktioniert auch der Abakus, eine Rechenmaschine die man viele hundert Jahre verwendet hat, und die in Varianten auch heute noch im Zeitalter des Taschenrechners in einigen Regionen der Erde verwendet wird.

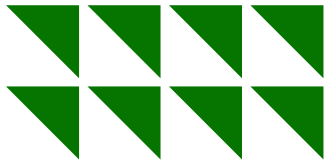
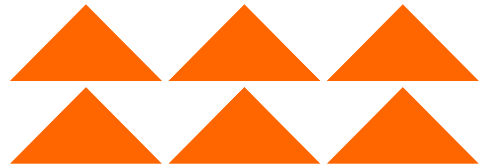
Stichwörter: Binäres Zahlensystem, Abakus

https://de.wikipedia.org/wiki/Dualsystem
https://de.wikipedia.org/wiki/Abakus_(Rechenhilfsmittel)

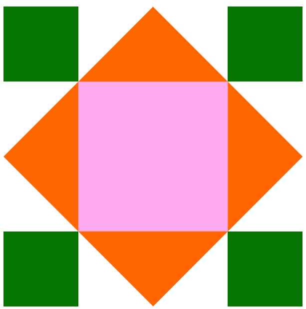
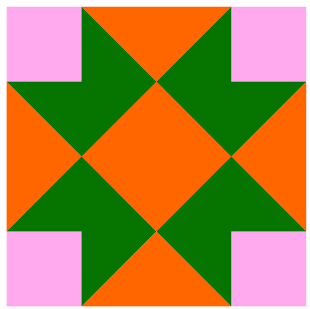
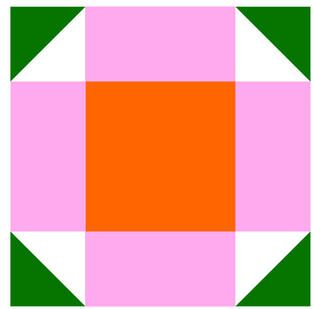
Beavercoins
2019-CH-03b

Rangoli ist eine Kunstform aus Indien. Dabei werden Muster auf den Boden gelegt. Diese Muster sind meist symmetrisch.

Priya hat für ihr Rangoli Steine in drei verschiedenen Formen: acht grüne Dreiecke, vier rosa Quadrate und sechs orange Dreiecke. Gleichfarbige Steine sind gleich gross:

Acht grüne Dreiecke	Vier rosa Quadrate	Sechs orange Dreiecke
		

Auf einer Webseite findet sie die folgenden Vorschläge für Rangoli (die weissen Flächen bleiben frei):

Vorschlag 1	Vorschlag 2	Vorschlag 3
		

Welche der drei Vorschläge für Rangoli kann Priya mit ihren Steinen legen?

A) Nur den Vorschlag 1.

B) Nur den Vorschlag 2.

C) Nur den Vorschlag 3.

D) Alle drei Vorschläge.

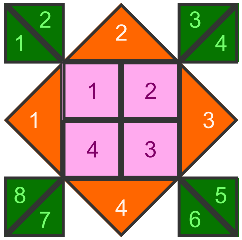
3/4	–
5/6	mittel
7/8	leicht
9/10	leicht
11–13	–

Rangoli
2019-IN-09

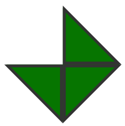
Lösung

Priya kann A) nur den Vorschlag 1 mit ihren Steinen legen.

Die folgende Graphik zählt die verschiedenartigen Steine im Vorschlag 1. Da sie von jedem Typ höchstens so viele Steine benötigt, wie sie zur Verfügung hat, kann Sie den Vorschlag 1 legen:

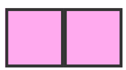


Für den Vorschlag 2 würde sie insgesamt zwölf grüne Dreiecke benötigen, denn eine der vier grünen Figuren im Vorschlag 2 benötigt jeweils drei grüne Dreiecke:



Priya hat aber nur acht grüne Dreiecke zur Verfügung, daher kann sie den Vorschlag 2 nicht legen.

Für den Vorschlag 3 würde sie insgesamt acht rosa Quadrate benötigen, denn eine der vier rosa Figuren im Vorschlag 3 benötigt jeweils zwei rosa Quadrate:



Priya hat aber nur vier rosa Quadrate zur Verfügung, daher kann sie den Vorschlag 3 nicht legen.

Da sie weder den Vorschlag 2 noch den Vorschlag 3 legen kann, kann auch die Antwort D) nicht richtig sein.

Das ist Informatik!

Rangoli ist eine Kunstform, die in Indien traditionell aus gefärbtem Reis und Mehl, aber auch aus farbigem Sand oder Blüten erstellt wird. Rangoli haben vor allem dekorative Zwecke, werden aber auch mit regionalen Traditionen oder Familientraditionen und guten Wünschen verbunden. Auch einige religiöse Traditionen verbinden sich mit Rangoli.

In dieser Aufgabe musste man eine komplexe Form in kleinere Formen zerlegen, die man dann mit den vorhandenen Grundformen abgleichen konnte. Man weiss dann, wie viele von den Grundformen jeweils benötigt werden. Diesen Vorgang nennt man *Dekomposition*, er kommt in der Informatik häufig vor.

Die zerlegten Formen mit Grundformen abzugleichen nennt man *Pattern Matching* (engl. für *Musterzuordnung* oder *Musterabgleich*). In der Informatik ist Pattern Matching von grosser Bedeutung, wobei nicht nur nach graphischen Mustern gesucht wird, sondern auch beispielsweise nach Wörtern in Texten oder Dateinamen im Dateisystem, oder auch beim Vergleich von Erbgut-Sequenzen bei der Verbrechersuche.

Stichwörter: Dekomposition, Pattern Matching

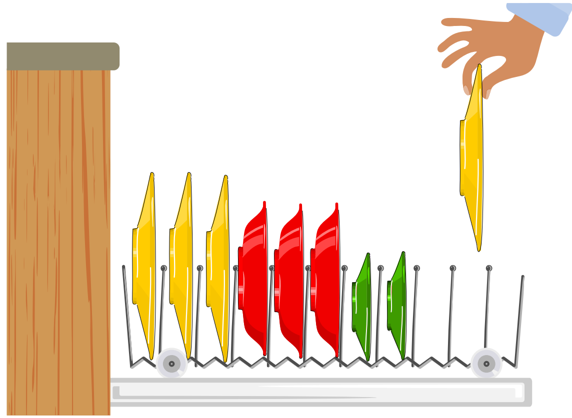
https://en.wikipedia.org/wiki/Rangoli
https://de.wikipedia.org/wiki/Pattern_Matching
https://en.wikipedia.org/wiki/Decomposition_(computer_science)

Rangoli
2019-IN-09

Abwaschmaschine einräumen

2019-RU-01

Urs ordnet seine Teller in der Abwaschmaschine, so dass ganz links die grossen Teller stehen, in der Mitte die Suppenteller und rechts die kleinen Teller. Zwischen den Tellern sind keine Lücken. Nach dem Nachtessen muss er einen weiteren grossen Teller in die Abwaschmaschine stellen. Er möchte beim Umstellen möglichst wenige Teller in der Abwaschmaschine anfassen, will die Ordnung aber beibehalten.



Wie viele Teller in der Abwaschmaschine muss er anfassen, damit der danach den grossen Teller an der richtigen Stelle einräumen kann?

A) 0

B) 1

C) 2

D) 3

E) 5

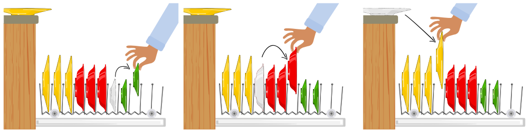
F) 8

3/4	5/6	7/8	9/10	11–13
–	–	leicht	–	–

Abwaschmaschine einräumen
2019-RU-01

Lösung

Am schnellsten ist Urs, wenn er den linken kleinen Teller nach rechts zur Seite stellt, den freigewordenen Platz mit dem linken Suppenteller auffüllt und den weiteren grossen Teller an den freigewordenen Platz stellt, so dass der neue Teller neu ganz rechts von allen grossen Tellern steht. Damit hat er zwei Teller in der Abwaschmaschine angefasst, die Antwort C) ist also richtig.



Es geht nicht schneller, denn der grosse Teller muss an einen Platz gestellt werden, an der ein grosser Teller oder der linke Suppenteller steht (es muss also mindestens ein Teller aus der Abwaschmaschine angefasst werden). Ausserdem muss der angefasste Teller wieder an einem Platz abgestellt werden: Wenn es ein grosser Teller ist, ist das Problem von neuem vorhanden und wenn es der linke Suppenteller ist, muss dieser wiederum an einen Platz gestellt werden, an dem ein Suppenteller oder der linke kleine Teller steht (es muss also mindestens ein zweiter Teller aus der Abwaschmaschine angefasst werden).

Das ist Informatik!

In dieser Aufgabe geht es letztlich darum, ein neues *Element* in eine bereits *sortierte Liste von Elementen* einzufügen. Solche Vorgänge kommen in Computern sehr häufig vor, so dass es sich lohnt Gedanken darüber zu machen, wie man dies effizient macht.

In dieser Aufgabe sind die „Kosten“ für das Bewegen eines einzelnen Tellers in der Abwaschmaschine relativ hoch. Das Erkennen hingegen, was für ein Typ Teller es jeweils ist, geht sehr schnell. Gleichzeitig sind viele gleichartige Teller in der Abwaschmaschine vorhanden. So lohnt es sich, für dieses Spezialproblem eine besondere Lösung mit dem Bewegen von nur zwei Elementen zu finden. Für Computer ist es in der Regel einfacher, einen richtige Platz für das Einfügen eines Elements in einer sortierten Liste zu finden und alle Elemente dahinter um jeweils einen Platz zu verschieben.

Diese Art von vergleichsbasiertem Sortieren nennt man dann auch *Sortieren durch Einfügen* (engl. *insertion sort*). Sie gehört zu den einfachen aber nicht sonderlich effizienten Sortierverfahren. Andere ähnliche Sortierverfahren sind *Sortieren durch Aufsteigen* (engl. *bubble sort*) oder *Sortieren durch Auswählen* (engl. *selection sort*). Das ebenfalls gängige *Quicksort*, das nach dem Prinzip *Teile und Herrsche* vorgeht, ist hingegen deutlich schneller, insbesondere für grosse Listen.

Stichwort: Sortieren

https://de.wikipedia.org/wiki/Sortierverfahren
https://de.wikipedia.org/wiki/Insertionsort
https://www.youtube.com/watch?v=ROalU379l3U

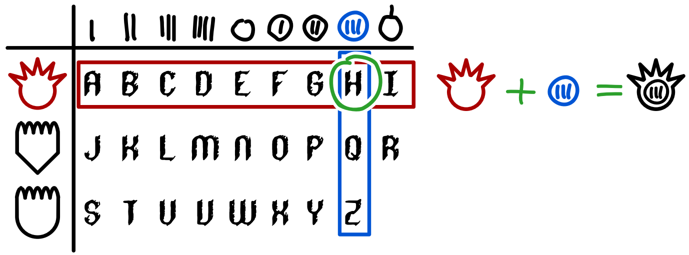
Nachricht der Urbiber

2019-CH-10

3/4	5/6	7/8	9/10	11–13
–	schwer	mittel	–	–

Ganz unten am Biberdamm findet die Biberin Dara ein uraltes Stück Holz. In das Holz sind unbekannte Zeichen eingeritzt. Dara nimmt an, dass dies eine Chiffrierungstabelle aus der Zeit ist, als die Urbiber den Biberdamm bewohnten.

Dara schaut die Tabelle lange an und glaubt zu wissen, wie sie funktioniert: Die unbekannten Zeichen sind eine Kombination der Symbole, die in den Spalten und Zeilen angegeben sind. Der Buchstabe „H" wäre damit so chiffriert:



Dara erinnert sich daran, dass sie an einer anderen Stelle im Biberdamm schon solche Zeichen gesehen hat. Tatsächlich steht dort:

Was bedeutet die Nachricht der Urbiber?

A) SAVEWATER

B) CLEARDAYS

C) SAVEMYDAM

D) CAREFORME

Nachricht der Urbiber

2019-CH-10

Lösung

Das erste Zeichen besteht aus den Formen und . Es ist also in der dritten Zeile und in der ersten Spalte zu finden: dort steht der Buchstabe S. Damit können nur noch die Antworten A) oder C) richtig sein.

Der zweite, dritte und vierte Buchstabe ist bei den Antworten A) und C) gleich. Die Zeichen entsprechen auch den Buchstaben A, V und E. Beim fünften Buchstaben aber unterscheiden sich die beiden Antworten wieder. Dieses Zeichen besteht aus den Formen und . Das entspricht dem Buchstaben W. Damit ist die Antwort A) richtig. Auch die letzten vier Zeichen entsprechen den Buchstaben A, T, E und R.

Es gibt eine Abkürzung, wie man die Aufgabe lösen kann. Wenn man anstelle des ersten Zeichens das letzte Zeichen anschaut, sieht man, dass sich alle Antworten im letzten Buchstaben unterscheiden. Die Formen und des letzten Zeichens entsprechen dem Buchstaben R und nur die Antwort A) hat diesen Buchstaben am Ende.

Das ist Informatik!

Datensicherheit ist heutzutage eine wichtige gesellschaftliche Aufgabe. Eine der Methoden, Daten vor unbefugtem Lesen zu schützen, ist sie zu chiffrieren. Die Wissenschaft des Verschlüsselns von Informationen (die *Kryptographie*) ist schon mindestens 3500 Jahre alt. Eine der ältesten bekannten Methoden der Verschlüsselung basiert auf dem Ersetzen von Buchstaben durch andere Buchstaben oder Zeichen. Beim *Verschlüsseln* (manchmal auch *Chiffrieren*) wird ein *Klartext* mit Hilfe eines *Schlüssels* in einen *Geheimtext* verschlüsselt. Das Rekonstruieren des Klartextes aus dem Geheimtext mit Hilfe des Schlüssels nennt man *Entschlüsseln* (manchmal auch *Dechiffrieren*). Wenn man den Klartext eines Geheimtextes ohne Kenntnis des Schlüssels herausfindet, nennt man das *Entziffern*.

Das Verschlüsselungsverfahren dieser Aufgabe ist eine sogenannte *monoalphabetische Verschlüsselung*. Bei diesen Verfahren wird für jeden Buchstaben genau ein neues Zeichen ausgewählt. Häufig werden dazu Systeme verwendet, die man sich leicht merken kann. Das System aus dieser Aufgabe ähnelt dem Freimaurer-Alphabet. Kryptoanalytiker, die solche Texte entziffern, würden spezielle Techniken wie Häufigkeitsanalysen oder *n*-Gramme im Geheimtext benutzen, um die Zeichen den richtig entschlüsselten Buchstaben zuzuordnen. Dass dies bei monoalphabetischen Verschlüsselungen generell möglich ist, hat Edgar Allen Poe in seiner 1843 veröffentlichten Kurzgeschichte "The Gold-Bug" gezeigt.

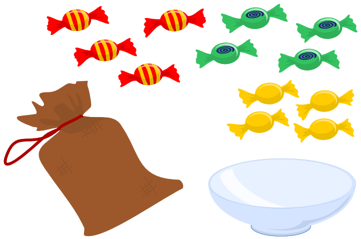
Was hätte Dara machen können, wenn sie die Tabelle nicht gehabt hätte, aber die vier möglichen Bedeutungen gekannt hätte? Sie hätte feststellen können, dass das zweite und sechste sowie das vierte und achte Zeichen gleich sind. Wenn sie von einer monoalphabetischen Verschlüsselung ausgeht, müsste sie nur noch den Text finden, dessen zweiter und sechster sowie vierter und achter Buchstabe gleich sind, und das wäre nur die Antwort A).

Stichwörter: Kryptographie, Monoalphabetische Verschlüsselung

https://de.wikipedia.org/wiki/Kryptographie
https://de.wikipedia.org/wiki/Monoalphabetische_Substitution
https://de.wikipedia.org/wiki/Freimaurer-Alphabet
https://de.wikipedia.org/wiki/Häufigkeitsanalyse
https://de.wikipedia.org/wiki/N-Gramm
https://de.wikipedia.org/wiki/Der_Goldkäfer
http://users.telenet.be/d.rijmenants/en/goldbug.htm

Ein Sack voller Bonbons

2019-HU-02



Petra hat in einem undurchsichtigen Sack vier rote, vier grüne und vier gelbe Bonbons. Zudem hat sie eine leere Schale.

Petra und Moritz spielen ein Spiel. Moritz darf während drei Runden ein Bonbon aus dem Sack ziehen. Für jedes gezogene Bonbon gelten folgende Regeln:

- Solange das gezogene Bonbon grün ist, legt er es in die Schale und er darf in dieser Runde ein weiteres Bonbon ziehen.
- Wenn das gezogenen Bonbon rot ist, legt es Moritz in die Schale und beendet die Runde.
- Wenn das gezogene Bonbon gelb ist, isst Moritz es direkt, ohne es in die Schale zu legen, und beendet die Runde.

Wie viele Bonbons hat Moritz am Ende des Spiels maximal in der Schale liegen?

A) 0	H) 7
B) 1	I) 8
C) 2	J) 9
D) 3	K) 10
E) 4	L) 11
F) 5	M) 12
G) 6	

3/4	–
5/6	–
7/8	schwer
9/10	mittel
11–13	leicht

Ein Sack voller Bonbons

2019-HU-02

Lösung

Die richtige Antwort ist H) 7.

Im günstigsten Fall werden alle vier grünen Bonbons gezogen. Das bedeutet, dass zum einen die vier grünen Bonbons in der Schale liegen und zum anderen, dass Moritz im Laufe der drei Runden vier Mal ein weiteres Bonbon ziehen durfte, also insgesamt sieben.

Für die restlichen drei Bonbons zieht Moritz im günstigsten Fall jeweils ein rotes Bonbon, die dann ebenfalls am Ende in der Schale liegen. Das macht dann insgesamt vier grüne und drei rote Bonbons, es liegen also sieben Bonbons in der Schale.

Mehr als sieben Bonbons können es nicht sein. Nach jedem Zug kommt höchstens ein Bonbon in die Schale und da es nur vier grüne Bonbons gibt, bei denen man ein weiteres Bonbon ziehen kann, sind es maximal sieben Bonbons.

Die Reihenfolge, in der die Bonbons im günstigsten Fall gezogen werden, ist relativ egal, solange das letzte gezogene Bonbon ein rotes ist, denn dann kann man durch die grünen Bonbons immer noch ein weiteres ziehen.

Das ist Informatik!

Zwei der drei Regeln der Aufgabe sind als *Verzweigungen* formuliert: *wenn* eine bestimmte Bedingung zutrifft, *dann* wird eine bestimmte Aktion ausgeführt. Solche Verzweigungen kommen beim Programmieren sehr häufig vor. Häufig werden hierfür die englischsprachigen Schlüsselwörter *if* (engl. für „wenn“) und *then* (engl. für „dann“) verwendet. Eine der Regeln ist so formuliert, dass etwas *solange* wiederholt wird, *bis* eine bestimmte Bedingung nicht mehr stimmt. So etwas nennt man eine *Schleife*, für die häufig das englische Schlüsselwort *while* (engl. für „solange“) verwendet wird. Solche Schleifen können auch als *Zählschleife* formuliert sein, die eine bestimmte Anzahl Wiederholungen vorgibt.

Man könnte also das Spiel von Petra auch so formulieren:

setze Runden auf 3
solange noch mindestens eine Runde vorhanden ist:
  verringere Runden um 1
  ziehe ein Bonbon
  solange das Bonbon grün ist, lege es in die Schale und
    ziehe ein Bonbon
  wenn das Bonbon rot ist, *dann* lege es in die Schale
  wenn das Bonbon gelb ist, *dann* iss es

Um die Aufgabe zu lösen, muss man das Programm *analysieren*. In einem so einfachen Fall wie diesem Programm könnte man natürlich einfach alle möglichen Reihenfolgen von Bonbons ausprobieren. Dies könnte sogar von einem Computer automatisiert durchgeführt werden. Die in der Lösung gelieferte Erklärung hingegen basiert darauf, die Zusammenhänge zu verstehen und so zu beweisen, dass ein bestimmtes Ergebnis wahr ist, ohne dass das Programm ausgeführt wird. Solche Analysen sind, wie die *Berechenbarkeitstheorie* zeigen konnte, nicht in jedem Fall von einem Computer durchführbar. Donald Knuth, einer der grossen Informatiker des 20. Jahrhunderts hat es mal so auf den Punkt gebracht: *„Vorsicht vor Fehlern im Code; ich habe nur bewiesen, dass er korrekt ist, ich habe ihn nicht ausprobiert“.*

Stichwörter: Verzweigung, Schleife, Berechenbarkeitstheorie

https://de.wikipedia.org/wiki/Bedingte_Anweisung_und_Verzweigung
https://de.wikipedia.org/wiki/Schleife_(Programmierung)
https://de.wikipedia.org/wiki/Berechenbarkeitstheorie
https://en.wikiquote.org/wiki/Donald_Knuth

Erdbeerklau

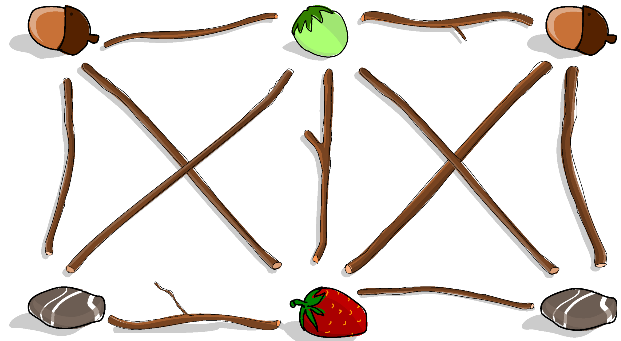
2021-CH-04c

3/4	5/6	7/8	9/10	11–13
–	schwer	mittel	–	–

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.

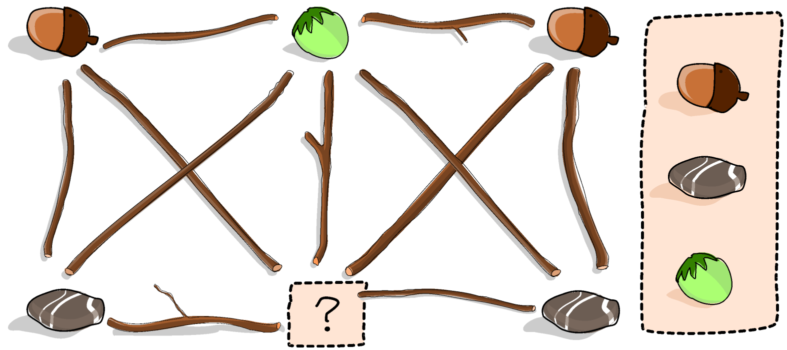
Hier ist das fertige Kunstwerk:



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.

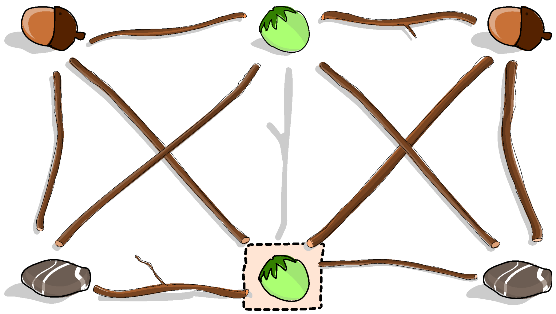


Erdbeerklau

2021-CH-04c

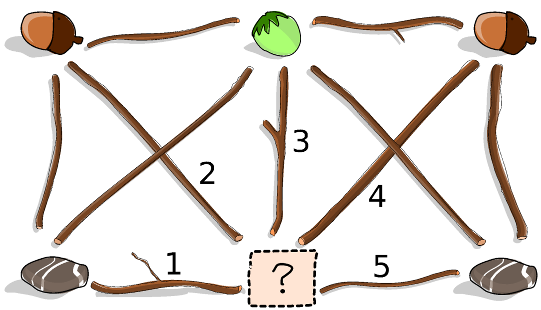
Lösung

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.



Das 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: Färbeproblem, minimale Kantenfärbung, Kantenfärbung, Clique

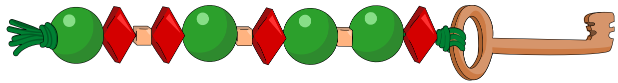
https://de.wikipedia.org/wiki/Färbung_(Graphentheorie)

https://de.wikipedia.org/wiki/Kantenfärbung

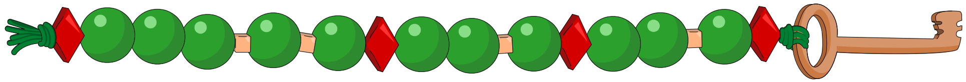
https://de.wikipedia.org/wiki/Clique_(Graphentheorie)

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: .

ANNA

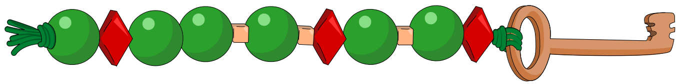


BELLA

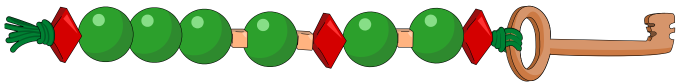


Welchen Schlüsselanhänger hat LENA gemacht?

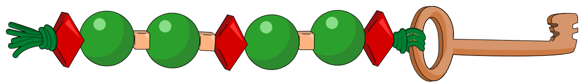
A)



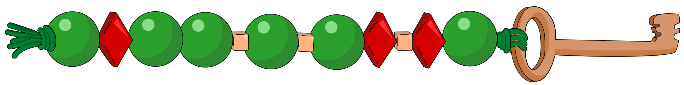
B)



C)



D)



3/4 mittel

5/6 leicht

7/8 –

9/10 –

11–13 –

Schlüsselanhänger

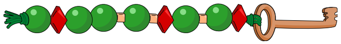
2021-SK-01

Schlüsselanhänger

2021-SK-01

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.

Das 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 Schreibtelegrafen. 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 Telegrafenleitungen 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: Zeichenkodierung, Morsecode

https://de.wikipedia.org/wiki/Zeichenkodierung

https://de.wikipedia.org/wiki/Morsecode

Dotties

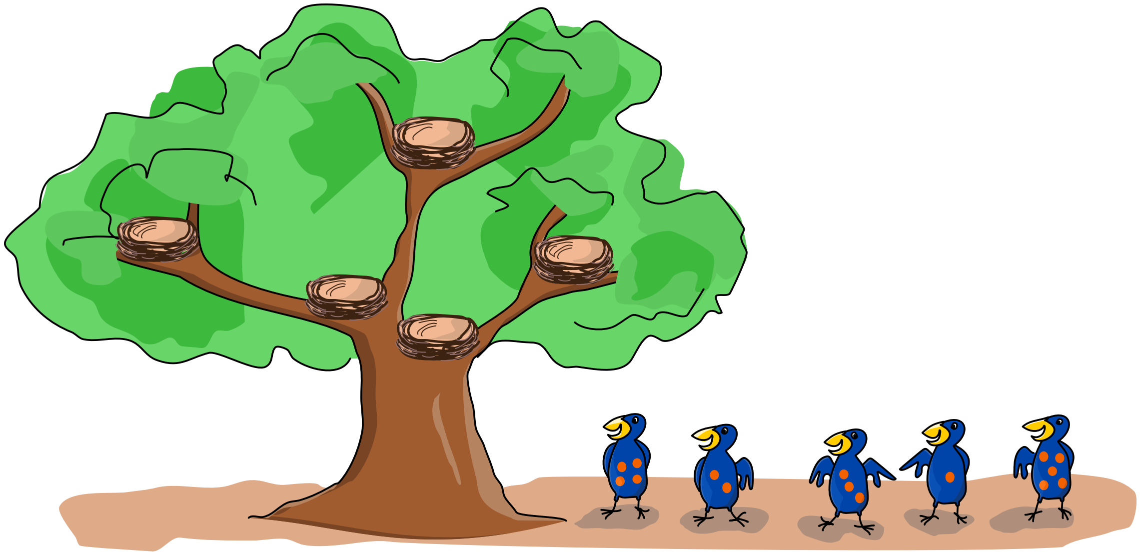
2021-CA-01b

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.

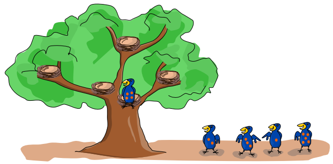
3/4	5/6	7/8	9/10	11–13
–	schwer	mittel	–	–

Dotties

2021-CA-01b

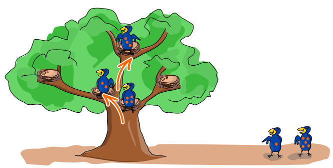
Lösung

So kommt man auf die richtige Lösung: Der erste Dotti, der mit 4 Punkten, zieht in das unterste Nest und bleibt dort.

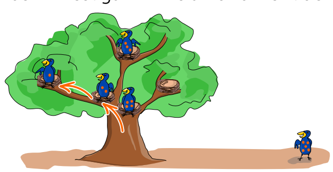


Der zweite Dotti hat 2 Punkte. Im untersten Nest sitzt der erste Dotti mit 4 Punkten. Weil 4 mehr als 2 ist, klettert der zweite Dotti nach links weiter und zieht in das erste freie Nest.

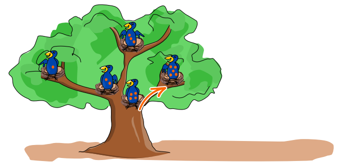
Der dritte Dotti hat 3 Punkte. Dieser klettert beim untersten Nest, wo der Dotti mit 4 Punkten sitzt, nach links, weil 4 mehr als 3 ist. Im nächsten Nest sitzt der Dotti mit 2 Punkten. Weil 3 mehr als 2 ist, klettert der dritte Dotti nach rechts weiter. Er zieht dann in das nächste freie Nest. Das ist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



Das ist Informaik!

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.

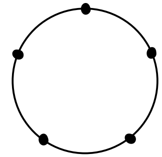
Es gibt viele Arten, auf die man Daten organisieren kann; man spricht von verschiedenen *Datenstrukturen*. Die Datenstruktur in dieser Biberaufgabe ist ein *binärer Suchbaum*. Das Wort *"binär"* kommt vom lateinischen Wort "bis" für "zweimal". Denn am Ende eines Astes (dort, wo in der Aufgabe ein Nest 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.*

Sichwörter: binärer Suchbaum

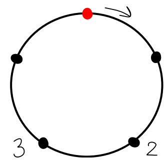
https://de.wikipedia.org/wiki/Binärer_Suchbaum

Stella liebt Sterne. Sie hat eine Methode erfunden, wie sie Sterne mit nur 2 Zahlen beschreiben kann, zum Beispiel «5:2»:

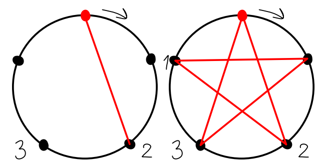
Die erste Zahl (5) gibt an, wie viele Spitzen der Stern hat. Stella zeichnet die Spitzen als Punkte auf einen Kreis.



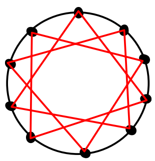
Die zweite Zahl (2) gibt an, wie Stella die Spitzen verbindet. Sie beginnt an einer Spitze und zählt die nächsten Punkte.



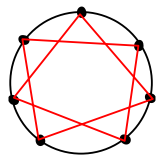
Ist die Zahl eine 1, verbindet sie immer die Spitzen direkt nebeneinander, bis alle Spitzen verbunden sind. Ist die Zahl eine 2, verbindet sie immer mit der übernächsten Spitze, u. s. w.



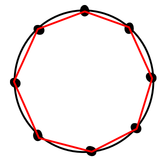
Stella hat diese 3 Sterne gezeichnet:



10:3

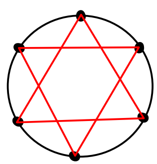


7:2



8:1

Wie würde Stella diesen Stern beschreiben?



A) 5:3

B) 6:2

C) 6:3

D) 7:2

3/4 –

5/6 schwer

7/8 mittel

9/10 leicht

11–13 –

Stellas Sterne

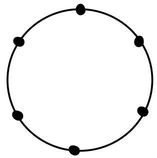
2015-SE-01

Stellas Sterne

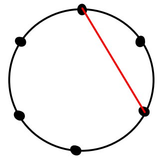
2015-SE-01

Lösung

Antwort B) ist richtig: 6:2.



Der Stern hat sechs Spitzen, daher "6".



Die Verbindungslinien führen immer zur übernächsten Spitze, das heisst zu jeder zweiten, daher "2".

Das ist Informatik!

Computer benötigen einfache und eindeutige *Repräsentationen* der Objekte (alternative Darstellung von Objekten), die sie verarbeiten. Bei Stellas System zum Sterne Zeichnen genügt die Anzahl der Spitzen, dazu kommt eine Vorschrift für die Verbindungslinien und schon ist die Form des Sterns präzise beschrieben. Farbe, Grösse und Position könnte man ebenso einfach beschreiben. In Vektorgrafikprogrammen wird als Repräsentation einer Grafik nicht Pixel für Pixel das eigentliche Bild gespeichert, sondern stattdessen eine Vorschrift zur geometrischen Konstruktion der Grafik. Das spart in der Regel Speicherplatz. Ausserdem ist es dann leicht möglich, durch Abändern weniger Zahlen in der Konstruktionsvorschrift die Grafik zu verändern, also z. B. zu vergrössern oder zu verkleinern.

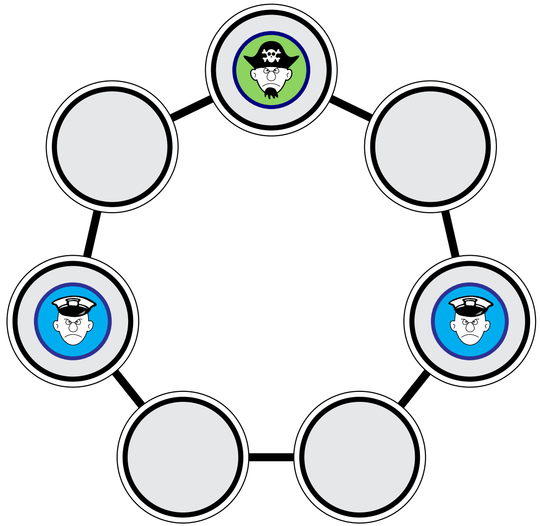
Stichworte: Sternpolygon, Schläfli-Stern, Vektorgrafik

https://de.wikipedia.org/wiki/Stern_(Geometrie)

Das Spiel “Piratenjagd” geht so: “Polizei” und “Pirat” ziehen abwechselnd. Ist die Polizei am Zug, muss sich einer der Polizisten auf ein benachbartes, freies Feld bewegen. Ist der Pirat am Zug, bewegt er sich um zwei Felder weiter. Das Spiel ist zu Ende, wenn der Pirat gezwungen ist, sich auf ein Feld zu bewegen, das von einem Polizisten besetzt ist.

Wenn der Pirat am Zug ist und das Spiel in der abgebildeten Situation ist, hat der Pirat also verloren – und die Polizei hat gewonnen. Die Polizei versucht also, den Piraten in diese Position zu zwingen.

Das Spiel beginnt in der Situation, die das Bild zeigt – aber die Polizei ist am Zug.



Nimm an, dass der Pirat keine Fehler macht. Hat die Polizei dann eine Chance zu gewinnen?

A) Die Polizei kann in 2 Zügen gewinnen.

B) Die Polizei kann in 3 Zügen gewinnen.

C) Die Polizei kann in 5 Zügen gewinnen.

D) Die Polizei hat keine Chance zu gewinnen.

Piratenjagd

2015-SI-07

3/4 –

5/6 –

7/8 –

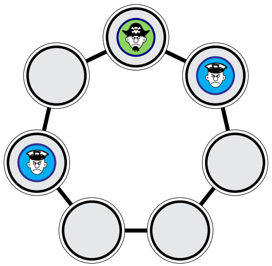
9/10 schwer

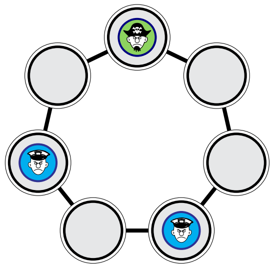
11–13 schwer

Lösung

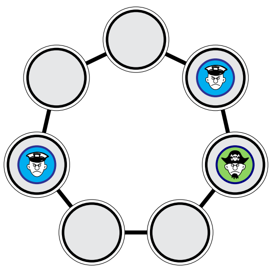
Antwort D ist richtig: Die Polizei hat keine Chance zu gewinnen.

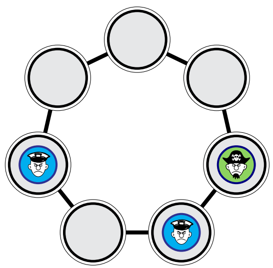
Nehmen wir an, das Spiel ist in der gezeigten Situation, und der Pirat ist am Zug – dann gewinnt die Polizei. Mit welchem Zug hat die Polizei den Piraten in diese (aus ihrer Sicht) Gewinnsituation gezwungen? Einer der Polizisten muss sich um ein Feld bewegt haben, nach oben oder unten. Nehmen wir an, es war der rechte Polizist; weil das Spielfeld symmetrisch ist, ist das keine Einschränkung. Vor dem Zug war das Spiel dann in einer der folgenden Situationen:





Welchen Zug kann der Pirat wiederum davor gemacht haben? Er muss von rechts gekommen sein (links steht ein Polizist). Also war das Spiel vor seinem Zug in einer dieser Situationen:





Nur aus einer dieser Situationen (oder einer der "gespiegelten" Situationen, die entstehen, falls in der vorletzten Situation der linke Polizist gezogen ist) kann es also zur Gewinnsituation der Polizei kommen. Weil der Pirat aber keine Fehler macht, wird er sich in diesen Situationen nicht nach oben bewegen, sondern nach links. Es kann also nicht zur Gewinnsituation kommen, und die Polizei hat keine Chance zu gewinnen.

Das ist Informatik!

Es gibt sehr viele Spiele mit zwei Spielern, z.B. Schach oder Dame. Viele dieser Spiele kann man auch gegen den Computer spielen. Programme für diese Spiele berechnen ihre eigenen Züge, indem sie von der aktuellen Situation ausgehen und die möglichen Züge berechnen, die sie selbst und ihr Gegner in der Folge machen können. Mit Hilfe von *Algorithmen* wie *Minimax* bewerten sie ihre eigenen Züge und nehmen dabei an, dass der Gegner keinen Fehler macht – wie hier der Pirat. Sind die Spiele sehr kompliziert (wie etwa Schach), können nicht alle Züge vorausberechnet werden; dann muss sich das Programm bei der Bewertung der eigenen Zugmöglichkeiten mit Abschätzungen helfen. Bei einigen Zwei-Personen-Spielen sind Programme besser als alle Menschen, z.B. im Schach, während bei anderen solchen Spielen wie etwa Go Menschen (noch) überlegen sind.

Stichwörter: Graphentheorie, Minimax-Algorithmus

https://de.wikipedia.org/wiki/Minimax-Algorithmus

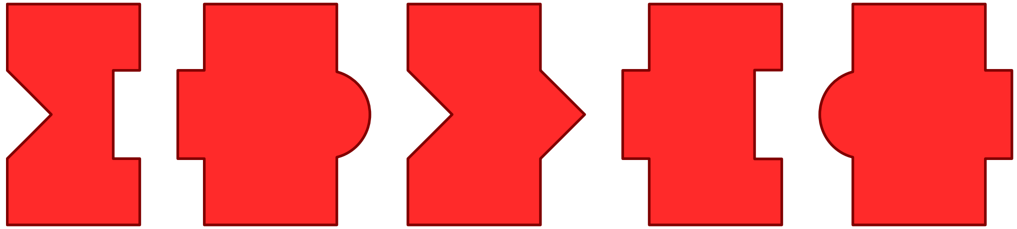
Piratenjagd
2015-SI-07

Passende Paare

2016-DE-09

Die Biber haben ein neues Spiel bestehend aus fünf Puzzleteilen. Einige dieser Puzzleteile können mittels passender Steckverbindungen zu Pärchen zusammengefügt werden.

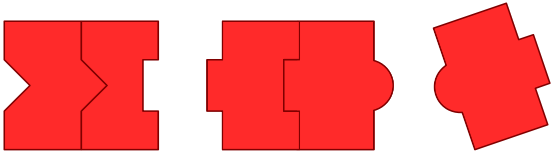
Erstelle so viele Pärchen wie möglich! Beachte, dass das Zusammenfügen nur funktioniert, wenn die Steckverbindungen die richtige Form haben.



3/4	mittel
5/6	leicht
7/8	–
9/10	–
11–13	–

Lösung

Diese zwei Pärchen können gleichzeitig erzeugt werden:



Fügt man die Puzzleteile anders zusammen, so kann man maximal ein Pärchen gleichzeitig erzeugen und es bleiben jeweils drei Puzzleteile übrig.

Das ist Informatik!

Wie hast Du diese Aufgabe gelöst? Vermutlich hat es ausgereicht, dass Du einfach mal etwas ausprobiert hast und dann festgestellt hast, dass es nicht mehr Lösungen gibt.

Bei komplizierteren Aufgaben, wenn man Computer einsetzt, kann man auch manchmal mit Ausprobieren aller Möglichkeiten zu einer Lösung kommen. Solange die Aufgabe nicht zu gross ist, geht das noch. Bei grösseren Aufgaben kann es sein, dass sogar ein Computer viele Jahre brauchen würde um sie zu lösen. In diesem Fall kommt oft eine „heuristische Suche“ zum Einsatz. Dabei werden dann nicht alle Lösungen ausprobiert, sondern es wird mithilfe von einfachen Regeln entschieden, in welche Richtung weiter gesucht wird und in welche Richtung nicht.

Stichwörter: Brute-Force-Methode, Heuristik, Greedy-Algorithmus

https://de.wikipedia.org/wiki/Brute-Force-Methode

https://de.wikipedia.org/wiki/Heuristik#Informatik

https://de.wikipedia.org/wiki/Greedy-Algorithmus

Passende Paare

2016-DE-09

Binärer Geburtstag

2016-CH-05a

3/4 –

5/6 mittel

7/8 leicht

9/10 –

11–13 –

Heute ist Bennos elfter Geburtstag. Bennos Mutter findet aber nur noch fünf Kerzen. Zum Glück weiss sie, wie sie die Zahl elf mit fünf Kerzen darstellen kann. Sie steckt sie alle nebeneinander auf den Kuchen:

- Die Kerze ganz rechts ist 1 wert.
- Alle anderen Kerzen sind das doppelte der Kerze rechts daneben wert.
- Die Werte aller brennenden Kerze werden addiert.

Zum Beispiel:

1

2

4

2 + 1 = 3

16 + 4 + 1 = 21

Welche Kerzen brennen an Bennos elftem Geburtstag?

A)

B)

C)

D)

E)

Lösung

Richtig ist: A) (01011)

Die Kerzen für die Zahlenwerte 8, 2 und 1 brennen: $0 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 = 11$.

Falsch sind:

B) 

(01110), denn die Kerzen für 8, 4 und 2 brennen: $0 \cdot 16 + 1 \cdot 8 + 1 \cdot 4 + 1 \cdot 2 + 0 \cdot 1 = 14$.

C) 

(10000), denn nur die Kerze für den Zahlenwert 16 brennt: $1 \cdot 16 + 0 \cdot 8 + 0 \cdot 4 + 0 \cdot 2 + 0 \cdot 1 = 16$.

D) 

(11010), denn die Kerzen für 16, 8 und 2 brennen: $1 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 0 \cdot 1 = 26$.

E) 

(11111), denn alle Kerzen brennen: $1 \cdot 16 + 1 \cdot 8 + 1 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 = 31$.

Das ist Informatik!

Jede beliebige Zahl kann in dieser Binärform dargestellt werden. Mit „Kerze an" oder „Kerze aus" wird ausgedrückt, ob ein Zahlenwert addiert wird oder nicht. Durch die Position der Kerze wird die Höhe des Zahlenwerts festgelegt. Das gleiche kann man auch mit Einsen (Kerze an) und Nullen (Kerze aus) ausdrücken. Das *Binärsystem* (auch *Dualsystem* genannt) wird intern von fast allen modernen Computern genutzt. Das hat praktische Gründe. Logische Schaltkreise für das Binärsystem sind einfacher zu realisieren als z. B. für das Dezimalsystem.

Stichworte: Binärsystem, Binäre Informationsdarstellung, Dualsystem

https://de.wikipedia.org/wiki/Dualsystem

Blumen und Sonnen

2016-CH-12

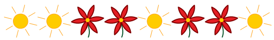
Barbara hat 2 Stempel bekommen. Einer druckt eine Blume, der andere eine Sonne. Sie überlegt, wie sie nur mit Blumen und Sonnen ihren Namen stempeln kann.

Für verschiedene Buchstaben bestimmt sie verschiedene Folgen von Blumen und Sonnen:

Buchstabe	**B**	**A**	**R**	**E**	**Y**
Folge					

Ihren eigenen Namen „Barbara“ muss sie dann so stempeln:

Nun stempelt Barbara den Namen eines ihrer Freunde:



Welchen Namen hat sie gestempelt?

A) Abby

B) Arya

C) Barry

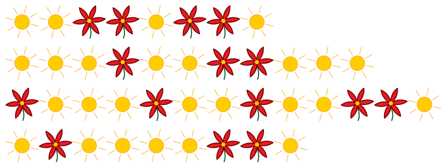
D) Ray

3/4	–
5/6	schwer
7/8	leicht
9/10	–
11–13	–

Lösung

Die richtige Antwort ist A) Abby. Die Namen von Barbaras Freunden haben folgende Codes:

Abby
Arya
Barry
Ray



Das ist Informatik!

Das Codieren von Daten kann auf verschiedene Weise geschehen. Zum Beispiel ist es oft üblich, dass die Zeichen, die auf der Tastatur eingegeben werden, in UTF-8, einer Variante von *Unicode*, abgespeichert werden. Dabei benötigt die häufigen Zeichen genau 1 *Byte*, das über 250 verschiedene Zeichen ermöglicht. Für seltenere Zeichen werden dann vier Byte Platz verwendet; damit ermöglicht man dann viele Millionen verschiedene Zeichen der verschiedensten Sprachen der Erde.

Das System funktioniert schon sehr gut, aber auch bei den häufigen Zeichen werden einige viel häufiger verwendet als andere, beispielsweise das „E" oder das „N" anstelle vom „X" oder vom „Ö". Auch hierfür gibt es sinnvolle Codes, die dann mit gänzlich variabler Länge von Symbolen arbeiten.

Bei solchen Codes mit variabler Länge ist es sinnvoll, dass ein Code eines Zeichen niemals der Anfang eines Codes eines anderen Zeichen ist. Dadurch kann die Bedeutung der einzelnen Codewörter schnell und einfach erkannt werden. Solche Codes nennt man *Präfixcode*. Ein bekannter Präfixcode ist der Morsecode.

Wenn man nun einen möglichst platzsparenden Code haben will, dann muss man wissen wie häufig die einzelnen Zeichen vorkommen und kann mit der sogenannten *Huffman-Codierung* einen besonders platzsparenden Code berechnen. Jeder Huffman-Code ist auch ein Präfixcode.

Stichwörter: Präfixcode, Huffman-Code, Datenkompression

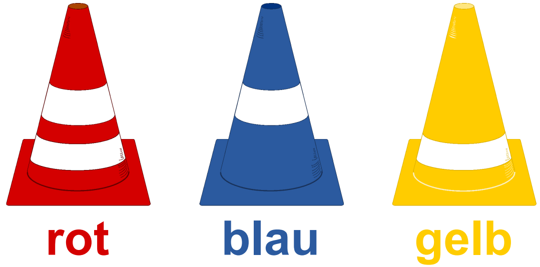
https://de.wikipedia.org/wiki/Präfixcode

https://de.wikipedia.org/wiki/Huffman-Kodierung

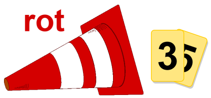
Verkehrshüetli-Versteck

2016-FR-02

Vreni versteckt Karten unter drei Verkehrskegeln. Dabei legt sie immer eine Karte unter einen bestimmten Kegel. Wenn dort schon eine Karte liegt, legt sie die neue Karte oben auf die alte Karte.



Welche Karte sie unter welchem Verkehrskegel versteckt hat, schreibt sie sich so auf:



rot ← 5

rot ← 3

Vreni hat sich notiert:

rot ← 3
gelb ← 5
rot ← 6
gelb ← 8
blau ← 1
gelb ← 3

Welche Karte liegt am Ende unter den jeweiligen Verkehrskegel zuoberst?

A) rot: 3, blau: 1, gelb: 5
B) rot: 9, blau: 1, gelb: 16
C) rot: 6, blau: 1, gelb: 3
D) rot: 8, blau: 1, gelb: 3

Lösung

Am Anfang liegt unter keinem Verkehrskegel eine Karte. Die folgende Tabelle zeigt an, welche Karten jeweils nach jedem Schritt unter den Verkehrskegeln liegen:

Befehl	rot	blau	gelb
	-	-	-
rot ← 5	5	-	-
gelb ← 5	5	-	5
rot ← 6	6	-	5
gelb ← 8	6	-	8
blau ← 1	6	1	8
gelb ← 3	6	1	3

Damit ist klar, dass die Antwort C) die richtige Lösung ist.

Das ist Informatik!

Ein Verkehrshüetli kann als eine *Variable* aufgefasst werden, deren Name „rot“, „blau“ oder „gelb“ ist. Eine Variable ist ein Ort im Speicher des Computers, an dem sich der Computer Werte merken kann, in diesem Fall sind es ganze Zahlen. Jeder Befehl ist eine *Zuweisung*, das heisst dass der bisherige Wert der gewählten Variablen durch einen neuen Wert überschrieben wird.

Zuweisungen werden in der Informatik häufig mit „=“ oder „:=“ notiert. Das führt jedoch häufig zu Denkfehlern, denn während in der Mathematik der Ausdruck „x = x + 1“ falsch ist, bedeutet er in der Informatik, dass man den Wert der Variablen nimmt, ihn um 1 erhöht, und das Ergebnis als neuen Wert in der Variablen speichert. Daher wird ausserhalb von Programmiersprachen oftmals der Pfeil „←“ für eine Zuweisung verwendet, denn „x ← x + 1“ ist nun auch für Mathematiker verständlich, während Informatiker sich nicht gross dran stören.

Stichwörter: Variable, Zuweisung

https://de.wikipedia.org/wiki/Variable_(Programmierung)

https://de.wikipedia.org/wiki/Zuweisung

Verkehrshüetli-Versteck

2016-FR-02

Gruppenarbeit

2016-IS-02

Sarah und ihre Mitschüler Alicia, Beat, Caroline, David und Emil sollen für den Unterricht eine Gruppenarbeit machen. Sie teilen sich die Arbeit auf, wobei Sarah die Aufgabe hat, die Ergebnisse zu sammeln. Das Ergebnis von Emil kann sie sich direkt geben lassen, aber um von den anderen die Ergebnisse zu bekommen, muss sie zunächst andere Ergebnisse vorzeigen:

- Um von David das Ergebnis zu bekommen, muss sie das Ergebnis von Alicia vorzeigen.
- Um von Beat das Ergebnis zu bekommen, muss sie das Ergebnis von Emil vorzeigen.
- Um von Caroline das Ergebnis zu bekommen, muss sie die Ergebnisse von Beat und David vorzeigen.
- Um von Alicia das Ergebnis zu bekommen, muss sie die Ergebnisse von Beat und Emil vorzeigen.

Ordne die Namen in eine Reihenfolge, bei der Sarah alle Ergebnisse erhalten kann:

Alicia → Beat → Caroline → David → Emil

3/4	5/6	7/8	9/10	11–13
–	–	–	leicht	–

Gruppenarbeit

2016-IS-02

Lösung

Die richtige Lösung ist:

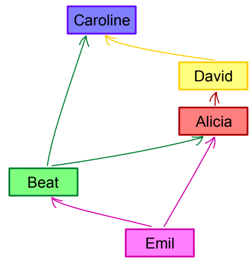
Emil → Beat → Alicia → David → Caroline

Emil ist der Einzige, der sein Ergebnis herausgibt, ohne dass er die Ergebnisse von den anderen sehen will, also muss er als erstes befragt werden. Beat ist der einzig übrige, der sein Ergebnis herausgibt, wenn ihm nur das Ergebnis von Emil gezeigt wird. Alicia ist dann die einzig übrige, die ihr Ergebnis herausgibt, wenn ihr nur die Ergebnisse von Emil und Beat gezeigt werden. David ist wiederum der einzig übrige, der sein Ergebnis herausgibt, wenn ihm nur die Ergebnisse von Emil, Beat und Alicia gezeigt werden (ihn interessiert dabei lediglich das Ergebnis von Alicia). Übrig bleibt Caroline, da man ihr aber die Ergebnisse von Beat und David zeigen kann, gibt auch sie ihr Ergebnis heraus.

Das ist Informatik!

Es passiert im Leben häufig, dass man bestimmte Dinge erst machen kann, wenn man bestimmte Voraussetzungen dafür erfüllt hat. Die Informatik beschäftigt sich mit solchen *Ablaufplanungen* (*Scheduling*) ebenfalls. Die Programme die hierfür entwickelt werden, werden beispielsweise in der Industrie eingesetzt. Die komplexen Abläufe wie den Bau eines Autos können so möglichst effizient, also zum Beispiel ohne unnötiges Warten auf Teile, gestaltet werden. Damit spart man viel Geld, weil man kein grosses Lager mehr benötigt und Maschinen möglichst viel arbeiten können.

Um einen Ablauf zu modellieren, erstellt man einen *Graph*, dessen *Knoten* die einzelnen Vorgänge sind (in diesem Fall das Abholen von Ergebnissen bei einer Person), und dessen *Kanten* durch Pfeile (sogenannte gerichtete Kanten, so dass ein *gerichteter Graph* entsteht) die Voraussetzungen darstellen. Ein solcher Graph für unser Problem sieht so aus:



Eine mögliche Lösung eines Ablaufplans ist ein Pfad durch den Graphen, der alle Knoten besucht. In unserem Graphen gibt es nur einen möglichen Pfad, nämlich die korrekte Lösung Emil → Beat → Alicia → David → Caroline.

Wie oben angedeutet, ist das Finden einer Lösung jedoch nur ein Teil des Problemlösens. Es gibt komplexere Situationen, wenn beispielsweise Dinge parallel stattfinden können. Zudem ist es häufig wichtig, eine Lösung zu finden, die beispielsweise die Kosten möglichst gering wählt, oder die besonders schnell sind. Oftmals jedoch ist es nicht praktikabel, einfach alle Lösungen auszuprobieren, so dass man sich mit *Algorithmen* zur Annäherung einer optimalen Lösung behilft.

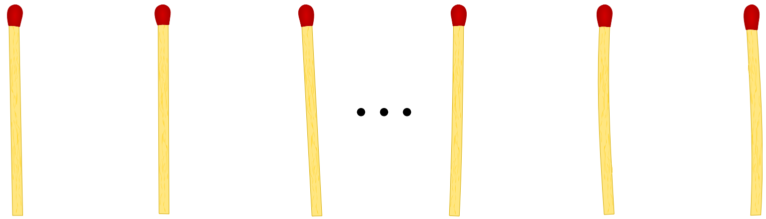
Stichwörter: Scheduling, Ablaufplan, gerichteter Graph

https://de.wikipedia.org/wiki/Scheduling

Nim

2016-CH-23

Beat und sein Freund spielen das Nim-Spiel. 13 Hölzchen liegen auf dem Tisch. Die beiden Spieler nehmen abwechselnd 1, 2 oder 3 Hölzchen weg. Wer das letzte Hölzchen nimmt, hat gewonnen.



Hinweis: Wenn noch vier Hölzchen auf dem Tisch liegen, kann Beat nicht mehr gewinnen. Diese Situation möchte er vermeiden.

Beat fängt an. Wie viele Hölzchen muss er wegnehmen, um das Spiel zu gewinnen?

A) 1

B) 2

C) 3

D) Das spielt keine Rolle.

3/4	–
5/6	schwer
7/8	mittel
9/10	leicht
11–13	–

Lösung

1 ist richtig. Dann bleiben 12 Hölzchen übrig. Der Freund nimmt 1, 2 oder 3 weg und Beat nimmt so viele, dass 8 übrig bleiben. Wieder nimmt der Freund 1, 2 oder 3 weg. Beat nimmt so viele, dass 4 übrig bleiben und der Freund nicht mehr gewinnen kann.

Wenn Beat 2 oder 3 Hölzchen nimmt, kann der Freund so reagieren, dass ein Vielfaches von 4 übrig bleibt. Dann kann Beat nicht mehr gewinnen.

Das ist Informatik!

In der *Spieltheorie* werden Spiele wie das *Nim-Spiel* oder das berühmte *Gefangenen-Dilemma* als Modelle verwendet, um strategische Probleme in der Wirklichkeit zu analysieren und Lösungswege zu finden. In einer Marktwirtschaft dienen die Erkenntnisse zum Beispiel dazu, Preise optimal zu gestalten. Eine Preissenkung kann zwar den Absatz erhöhen, aber sie verringert gleichzeitig den Gewinn pro verkauftem Produkt. Umgekehrt vergrössert eine Preissteigerung den Gewinn pro verkauftem Produkt, aber sie kann auch dazu führen, dass der Absatz und damit der Gesamtgewinn sinkt. Mit Modellen aus der Spieltheorie können mögliche Reaktionen der Käufer auf Preisänderungen vorhergesagt werden. Wie bedeutend diese Modelle für die Wirtschaft sind, sieht man daran, dass bereits mehrere Nobelpreise für spieltheoretische Arbeiten verliehen wurden.

Stichwörter: Nim-Spiel, Spieltheorie, Entscheidungsbaum

http://www.mathematische-basteleien.de/nimspiel.html

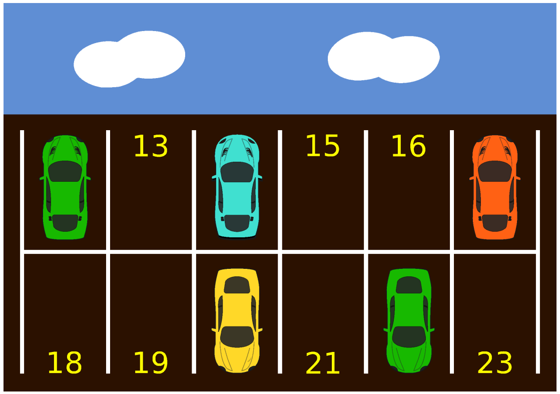
http://scienceblogs.de/zoonpolitikon/2008/04/22/spieltheorie-einfach-erklart-i-einleitung-und-gefangenendilemma/

Parkplatz

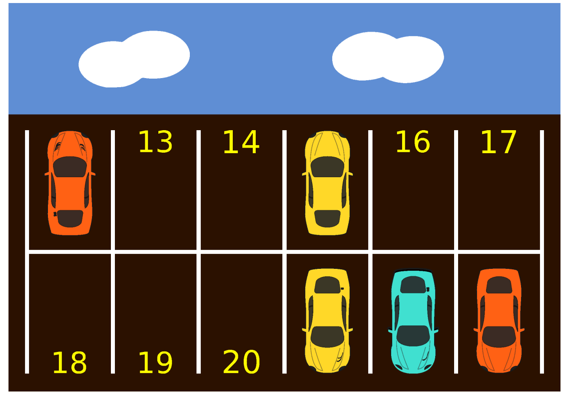
2017-CA-01

Auf dem Biberplatz hat es 12 Parkplätze für Autos. Jeder Parkplatz ist mit einer Nummer gekennzeichnet. Die Bilder unten zeigen, welche Parkplätze am Montag und welche am Dienstag besetzt waren.

Montag



Dienstag



Wie viele Parkplätze waren sowohl am Montag als auch am Dienstag frei?

A) 3 Parkplätze

B) 4 Parkplätze

C) 5 Parkplätze

D) 6 Parkplätze

3/4 leicht

5/6 leicht

7/8 –

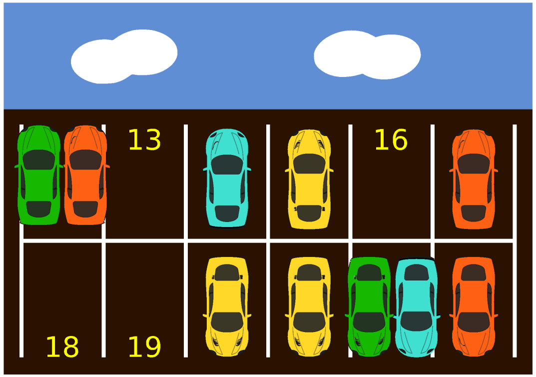
9/10 –

11–13 –

Lösung

Die Parkplätze 13, 16, 18 und 19 waren sowohl am Montag als auch am Dienstag frei.

Wenn die Autos von beiden Tagen auf einem Parkplatz dargestellt werden, sehen wir, welche Parkplätze frei bleiben. Diese Parklücken können dann einfach gezählt werden. Am Montag und Dienstag waren insgesamt 4 Parkplätze durchgehend frei.



Das ist Informatik!

Alle Arten von Daten können als eine Folge von Nullen und Einsen *codiert* werden. Jede Null oder Eins wird ein *Bit* genannt. Eine solche Sequenz heisst *Binärcode*.

Bei dieser Aufgabe können wir die Anwesenheit eines Autos auf einem Parkplatz mit einer Eins (1) und eine Abwesenheit als einer Null (0) codieren. So lässt sich die Belegung des Parkplatzes als Reihenfolge von Bits *modellieren*. Der Montag entspricht 101001001010, der Dienstag entspricht 100100000111. Mit Hilfe einer logischen *ODER (OR)* Verknüpfung werden alle Parkplätze, welche mindestens an einem der beiden Tage belegt waren mit einer 1 gekennzeichnet. Die richtige Antwort berechnen wir, indem wir die Reihen übereinander legen. Wir sehen, dass aus 101001001010 ODER (OR) 100100000111 schlussendlich 101101001111 wird.

Diese Binärzahl enthält 4 Nullen, also 4 freien Parkplätzen.

Stichwörter: Bits, Binär, ODER, logischer Operator

https://de.wikipedia.org/wiki/Disjunktion

Parkplatz
2017-CA-01

Fünf Hölzli

2017-SK-04

3/4 –

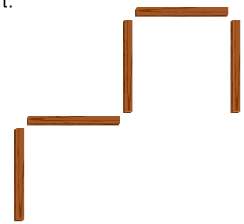
5/6 schwer

7/8 mittel

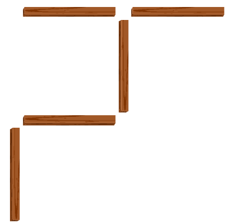
9/10 –

11–13 –

Fünf Hölzli liegen so auf dem Tisch:



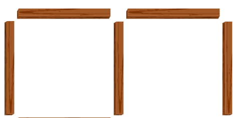
Nola nimmt ein Hölzli und legt es woanders hin. Jetzt liegen die Hölzli so:



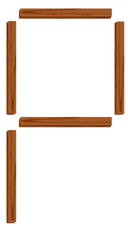
Danach nimmt Bert ein Hölzli und legt es woanders hin.

*Wie können die Hölzli jetzt **nicht** liegen?*

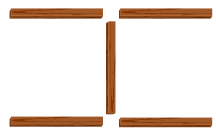
A)



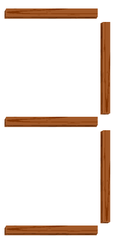
B)



C)



D)

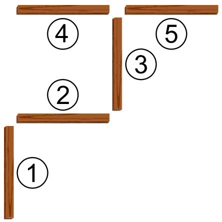


Fünf Hölzli

2017-SK-04

Lösung

Zum Erklären der Lösung werden die Hölzli durchnummeriert. Nach Nola liegen die Hölzli so:



Damit die Hölzli so wie in A) liegen, muss Bert das Hölzli 1 woanders hin legen.

Damit die Hölzli so wie in B) liegen, muss Bert das Hölzli 5 woanders hin legen.

Damit die Hölzli so wie in C) liegen, muss Bert das Hölzli 1 woanders hin legen.

Damit die Hölzli so wie in D) liegen, müsste Bert zwei Hölzli, nämlich die Hölzli 1 und 5 woanders hin legen. Er hat aber nur ein Hölzli woanders hin gelegt.

Das ist Informatik!

Um die Lage der Hölzchen zu verändern, haben Nola und Bert eine einfache Möglichkeit, nämlich genau ein Hölzchen woanders hin zu legen. Wenn ein Freund ihnen sagen möchte, wie sie vorgehen sollen, kann er eine entsprechende *Anweisung* benutzen: „Lege ein Hölzchen woanders hin!" Diese Anweisung ist jedoch nicht eindeutig. Er muss auch noch sagen, welches Hölzchen wohin muss.

Wer Computer programmiert, braucht dazu eindeutige Anweisungen. Gewisse Anweisungen kennt der Computer bereits, neue Anweisungen können aus bereits bestehenden zusammengesetzt werden.

Du kannst neue komplexeren Anweisungen erschaffen, indem du verschieden Anweisungen hintereinander aufschreibst oder indem du aufschreibt, dass eine Anweisung mehrfach hintereinander ausgeführt werden soll. Du kannst auch festlegen , dass eine Anweisung nur unter bestimmten *Bedingungen* ausgeführt werden soll. Das sind die drei wichtigsten Möglichkeiten, in Computerprogrammen aus (anfänglich einfachen) Anweisungen immer komplexere Anweisungen zu machen.

Stichwörter: Anweisungen, Zustandsveränderung, Programmieren

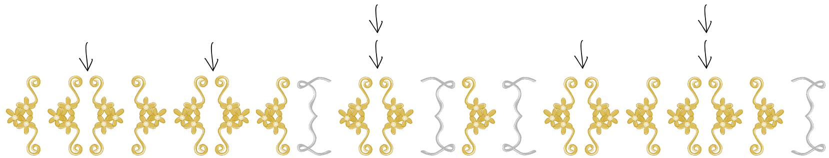
https://de.wikipedia.org/wiki/Anweisung_(Programmierung)

Klammeranhänger

2017-AT-06

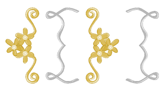
Familie Biber produziert selbstgemachte Anhänger für ein Mittelalterfest. Dazu verwenden sie klammernförmige Ornamente, die immer paarweise verwendet werden. Um einen Anhänger herzustellen, beginnt man mit einem der beiden folgenden Paare:

Danach werden wiederholt weitere Klammernpaare an einer beliebigen Stelle eingefügt, wie die drei Beispiele unten zeigen:

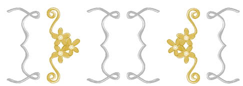


Welcher der folgenden Anhänger wurde mit der obigen Methode hergestellt?

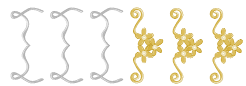
A)



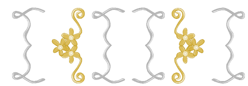
B)



C)



D)



3/4	–
5/6	schwierig
7/8	leicht
9/10	–
11–13	–

Klammeranhänger

2017-AT-06

Lösung

D) ist die richtige Antwort. In das ursprüngliche Klammernpaar wurde ein zusätzliches Klammernpaar eingefügt und in dieses wiederum ein weiteres.

Alle anderen Anhänger wurden nicht mit der beschriebenen Methode hergestellt:

- Von links beginnend ist der erste Fehler bei Position 3: eine Klammer wurde geschlossen, bevor die Klammer von Position 2 geschlossen wurde.
- Von links beginnend ist der erste Fehler schon bei Position 1: Eine Klammer wurde geschlossen, bevor sie geöffnet wurde
- Von links beginnend ist der erste Fehler bei Position 4: eine Klammer wurde geschlossen, bevor sie überhaupt geöffnet wurde.

Das ist Informatik!

Die Regeln zur Herstellung von Anhängern sind genau die gleichen Regeln wie für Klammernausdrücke in der Mathematik oder in der Informatik. Ausdrücke ohne Fehler nennt man *wohlgeformt*. Einen wohlgeformten Ausdruck nennt man auch *syntaktisch korrekt* in dem Sinne, dass sie die vorgegebenen syntaktischen Regeln, der Grammatik einer formalen Sprache, befolgen. *Syntaxfehler* sind in der Regel viel leichter zu beheben als Fehler, die zwar syntaktisch korrekt sind aber andere, mitunter subtile, Denkfehler enthalten. Die letzteren nennt man *semantische Fehler*.

Stichwörter: Wohlgeformtheit, Syntax, Semantik

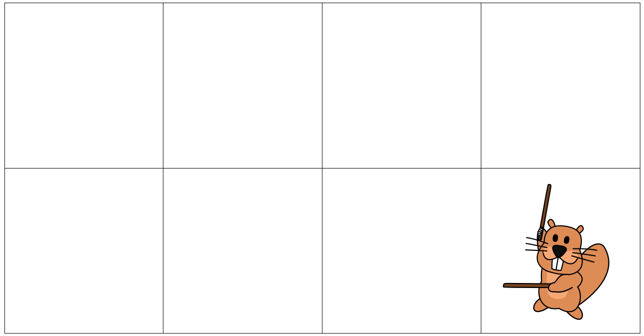
https://de.wikipedia.org/wiki/Syntax

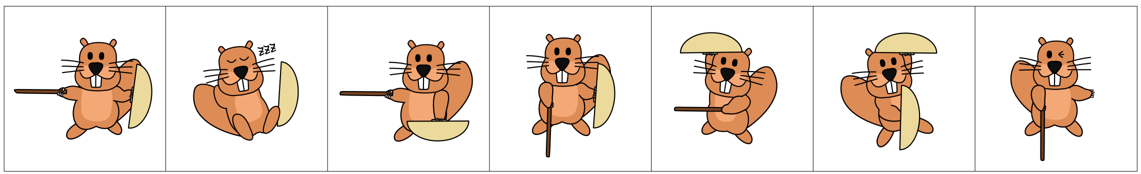
https://de.wikipedia.org/wiki/Semantik

Japanischer Stockkampf

2017-JP-02

Lucia und ihre Freunde sind Mitglieder eines Clubs für japanischen Stockkampf. Für ein Foto möchten sie sich auf dem Schulhof so aufstellen, dass jeder Stock auf ein Schild zeigt. Dafür wurden Felder auf den Schulhof gezeichnet. Lucia hat sich bereits in Pose gestellt, darunter sind die Lieblingsposen ihrer Freunde:





Ordne die Bilder der Freunde in die Felder auf dem Schulhof, so dass am Ende jeder Stock auf ein Schild zeigt.

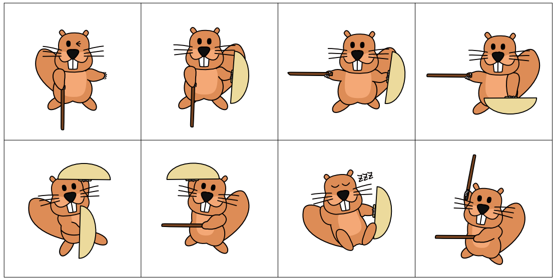
3/4	–
5/6	schwer
7/8	mittel
9/10	–
11–13	leicht

Japanischer Stockkampf

2017-JP-02

Lösung

So ist es richtig:



Die Bilder der Freunde müssen so in die Felder bewegt werden, wie es im Bild oben gezeigt wird. Dann zeigt jeder Stock auf ein Schild. Eine andere Möglichkeit, die Bilder so anzuordnen, dass diese Bedingung erfüllt ist, gibt es nicht.

Das ist Informatik!

Sieben Bilder müssen an die richtige Stelle geschoben werden. Wer versucht diese Aufgabe durch einfaches Ausprobieren zu lösen braucht viel Zeit. Denn es gibt $1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 \cdot 7 = 7! = 5040$ verschiedene Möglichkeiten, die sieben Bilder zu arrangieren. Die meisten davon sind natürlich falsch. Mit ein bisschen Logik findest du die Lösung schneller. Überlegen wir mal:

1. Alle Biber, die einen Stock oder ein Schild nach oben halten, müssen in der unteren Reihe stehen.
2. Alle Biber, die einen Stock oder ein Schild nach unten halten, müssen in der oberen Reihe stehen.
3. Es gibt nur einen einzigen Biber, der sein Schild nach unten hält und deshalb oberhalb von Lucia stehen kann.

Mit diesen Regeln kannst du den Suchraum für das Finden einer richtigen Lösung auf wenige Möglichkeiten eingrenzen. Ein Verfahren zum systematischem Ausprobieren aller Lösungsmöglichkeiten nach dem Prinzip von "Versuch und Irrtum" ist das *Backtracking*. Ein solches Verfahren ist aber nur dann schnell genug, wenn der Suchraum klein ist. Deshalb ist das Eingrenzen durch *logische Regeln* so wichtig.

Sichtwörter: Logisches Denken, Schlussfolgerung, Backtracking

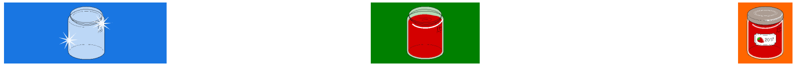
https://de.wikipedia.org/wiki/Backtracking

http://www.inf-schule.de/grenzen/komplexitaet/affenpuzzle/einstieg_affenpuzzle

Opas Marmelade

2017-RU-04

Weil ihr Opa Marmelade kocht, helfen Anna, Peter und Lisa, die Marmelade in Gläser abzufüllen. Dazu müssen sie jeweils diese Arbeitsschritte erledigen – und zwar genau in dieser Reihenfolge:



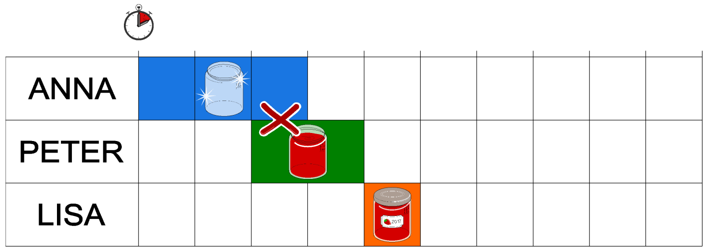
Ein Glas spülen dauert 3 Minuten.

Marmelade in ein Glas füllen dauert 2 Minuten.

Ein Glas zu verschliessen dauert 1 Minute.

Anna, Peter und Lisa wollen sich die Arbeit aufteilen und dazu einen Plan erstellen. Dabei müssen sie beachten: Ein Arbeitsschritt muss komplett erledigt sein, bevor der nächste drankommen kann. Ein Glas muss ganz sauber sein, bevor man Marmelade einfüllt. Und das Glas darf natürlich erst geschlossen werden, wenn es fertig gefüllt ist.

Der folgende Plan ist also nicht möglich:



Anna, Peter und Lisa wollen in 10 Minuten möglichst viele Gläser abfüllen. Erstelle dazu einen Plan, indem Du die Arbeitsschritte auf dem anfangs leeren Plan ordnest:

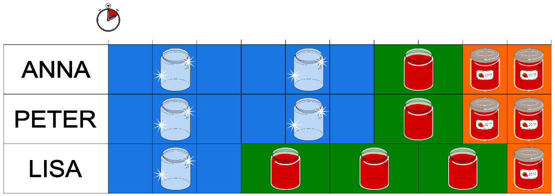
ANNA										
PETER										
LISA										

3/4 –
5/6 schwer
7/8 –
9/10 –
11–13 –

Opas Marmelade
2017-RU-04

Lösung

Hier ist ein Plan, mit dem Anna, Peter und Lisa fünf Gläser in 10 Minuten abfüllen können. Es gibt noch andere Pläne, mit denen sie das ebenfalls schaffen können. In diesen Plänen sind die drei Zeilen – also die Folgen von Arbeitsschritten, die die einzelnen Kinder erledigen sollen – gleich wie im abgebildeten Plan, aber anderen Kindern zugeordnet.



Mehr als fünf Gläser können die Kinder in 10 Minuten nicht abfüllen. Das Abfüllen eines Marmeladenglases dauert 6 Minuten; denn 3+2+1=6. Jedes der drei Kinder hat 10 Minuten Arbeitszeit. Somit beträgt die gesamte Arbeitszeit 30 Minuten. Es ist nicht möglich in 10 Minuten mehr als 5 Gläser fertig zu stellen; denn 30/6=5. Fünf Gläser sind also das theoretische Optimum. Es kann mit dem abgebildeten Plan erreicht werden.

Das ist Informatik!

In dieser Biberaufgabe arbeiten Anna, Peter und Lisa gleichzeitig, also parallel an den einzelnen Schritten eines Arbeitsvorgangs. Parallelverarbeitung gibt es auch in der Informatik. Zum Beispiel werden in Rechenzentren grosse Datenmengen von vielen Computern zur gleichen Zeit (parallel) verarbeitet. Dabei müssen – wie in der Aufgabe – einzelne Arbeitsschritte auf die Computer in sinnvoller Weise verteilt werden. Aber auch in einem einzigen Computer oder einem Smartphone gibt es mehrere Arbeitseinheiten (Prozessorkerne), die Arbeitsschritte parallel verarbeiten können. Die Erstellung eines Arbeitsplans (mit dem natürlich möglichst viel erledigt werden soll) nennt die Informatik auch *Scheduling*.

Aber auch ausserhalb der Informatik gibt es ähnliche Probleme. Für die Planung von Abläufen in einem grossen Projekt verwendet man die Netzplantechnik. In einem Netzplan ist festgelegt, in welcher Reihenfolge Arbeitsschritte ausgeführt werden und wie lange sie dauern. Die Aufstellung eines guten Netzplans ist wie Scheduling. Informatikkenntnisse helfen also auch, wenn man gar nicht als Informatikerin oder Informatiker arbeitet.

Stichwörter: Scheduling, Parallelverarbeitung

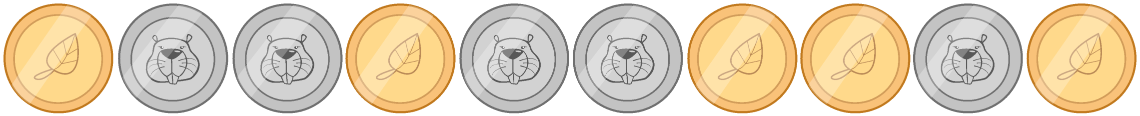
https://de.wikipedia.org/wiki/Netzplantechnik

Münzdreh

2017-US-03

Christine besitzt zehn Münzen, die auf der einen Seite golden () und auf der anderen Seite silbern () sind.

Sie legt die Münzen so auf den Tisch:



Wie häufig muss sie je zwei nebeneinanderliegende Münzen umdrehen, so dass am Ende alle goldfarbenen Seiten aufgedeckt sind?

A) 1 mal

B) 2 mal

C) 4 mal

D) 6 mal

E) 8 mal

F) Es ist nicht möglich.

3/4 –

5/6 –

7/8 mittel

9/10 leicht

11–13 –

Lösung

Die richtige Antwort ist F) Es ist nicht möglich.

Jedes Mal wenn sie zwei Münzen umdreht, bleibt die Anzahl der silbernen Seiten gleich. Es werden zwei silberne Seiten mehr oder zwei silberne Seiten weniger. Die Parität, also ob die Anzahl der silbernen Seiten gerade oder ungerade ist, bleibt immer gleich.

Die Parität der silbernen Seiten ist am Anfang ungerade … und sie bleibt immer ungerade. Eine Situation mit keiner silbernen Seite, was ja eine gerade Anzahl silberner Seiten wäre, kann also nie erreicht werden.

Das ist Informatik!

Paritäten können schnell und einfach berechnet werden. Man kann mit ihnen einfach überprüfen, ob eine Übertragung korrekt war (wie beispielsweise ein an der Kasse eingelesener Barcode) oder eine Zahl richtig eingegeben wurde (wie beispielsweise eine Kontonummer im Online-Banking). Wenn man etwas kompliziertere Berechnungen vornimmt, kann man sogar bestimmte Fehler korrigieren, ohne dass die Daten neu übertragen werden müssen.

Stichwörter: Parität, Paritätsbit

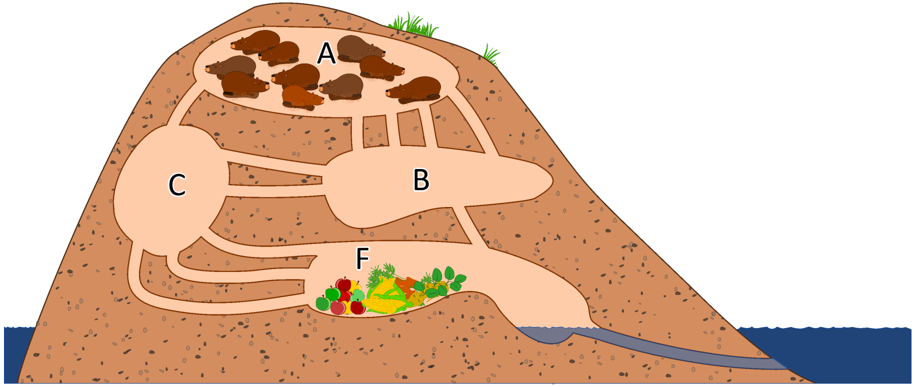
https://de.wikipedia.org/wiki/Paritätsbit

Münzdreh
2017-US-03

Gänge im Biberbau

2017-CH-07b

10 Biber befinden sich in Raum A und wollen möglichst rasch in den Raum F zum Fressen kommen. Ein Biber braucht 1 Minute um durch einen Gang zu laufen. Leider kann durch jeden Gang immer nur ein Biber gleichzeitig laufen. Sie können dabei also nicht direkt hintereinander durch den Gang laufen. In den Räumen A, B, C, F ist genug Platz für alle Biber und das Durchqueren eines Raums benötigt keine Zeit.



Nach wieviel Minuten können alle 10 Biber in Raum F sein? Gib die kürzest mögliche Zeit an!

3/4 –
5/6 –
7/8 –
9/10 schwer
11–13 mittel

Alle 10 Biber können bereits nach 4 Minuten in Raum F sein. Im Biberbau gibt es zwei kürzeste Wege zum Raum F. Beide Wege können innerhalb von 2 Minuten nur jeweils einen Biber ans Ziel bringen, in 3 Minuten sind es jeweils 2 Biber:

- A → B → F
- A → C → F

Der Weg A → B → C → F bietet Platz für 2 Biber, dauert aber 3 Minuten. Nach 3 Minuten haben wir insgesamt erst 6 Biber in Raum F. Wir brauchen noch eine 4. Minute. So können z.B. alle 10 Biber in 4 Minuten ans Ziel kommen:

Anzahl Biber im Raum	A	B	C	F
Situation zu Beginn	*10*	*0*	*0*	*0*
3 Biber gehen von A nach B (weniger als möglich ist)				
1 Biber geht von A nach C				
Situation nach 1 Minute	*6*	*3*	*1*	*0*
3 Biber gehen von A nach B (weniger als möglich ist)				
1 Biber geht von B nach F				
2 Biber gehen von B nach C				
1 Biber geht von C nach F				
1 Biber geht von A nach C				
Situation nach 2 Minuten	*2*	*3*	*3*	*2*
1 Biber geht von A nach B (kürzester Weg)				
1 Biber geht von B nach F				
2 Biber gehen von B nach C				
1 Biber geht von A nach C (kürzester Weg)				
3 Biber gehen von C nach F				
Situation nach 3 Minuten	*0*	*1*	*3*	*6*
1 Biber geht von B nach F				
3 Biber gehen von C nach F				
Situation nach 4 Minuten	***0***	***0***	***0***	***10***

Es gibt verschiedene Lösungswege, wie die Biber innerhalb von 4 Minuten in Raum F kommen. Bei der hier gezeigten Lösung müssen die Biber in keinem Raum auf das Weitergehen warten..

Das ist Informatik!

Das Netzwerk der Gänge kann als sogenanntes *Fluss-Netzwerk* aufgefasst werden. Die Anzahl der Gänge zwischen zwei Räumen bestimmt wie viele Biber innerhalb einer Minute von einem zum anderen Raum gehen können. Es ist die sogenannte *Kapazität* der Verbindung zwischen zwei Räumen, der den Fluss zwischen diesen Räumen begrenzt.

Ein Fluss-Netzwerk ist in der *Graphentheorie* ein *gerichteter Graph*, bei dem jede *Kante* eine Kapazität hat (die Anzahl der Gänge in unserer Aufgabe). Ein Fluss, der durch die Kanten des Graphen fliesst, ist durch die Kapazität der Kanten begrenzt. Mit Hilfe von Flüssen in Netzwerken kann ein Computernetzwerk oder ein Verkehrsnetzwerk simuliert werden und die Engpässe in den Verkehrsströmen aufgedeckt werden.

In Fluss-Netzwerken ist insbesondere der maximal mögliche Fluss zwischen zwei Knoten interessant. In unserer Aufgabe wären das 4 Biber, die pro Minute von Raum A zum Raum F ohne Wartezeit in einem der Räume laufen können. Der sogenannte *Ford-Fulkerson Algorithmus* kann so einen maximalen Fluss berechnen.

Stichwörter: Graph, Fluss in Netzwerken, Ford-Fulkerson Algorithmus

https://de.wikipedia.org/wiki/Flüsse_und_Schnitte_in_Netzwerken

https://de.wikipedia.org/wiki/Gerichteter_Graph

https://de.wikipedia.org/wiki/Algorithmus_von_Ford_und_Fulkerson

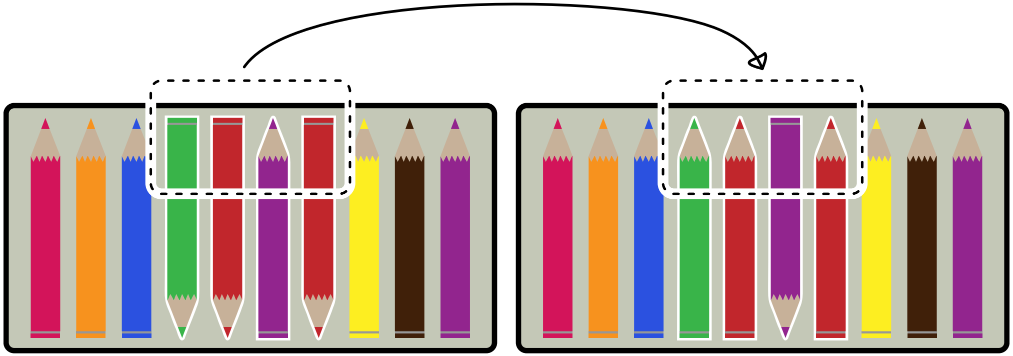
Adas Farbstifte

2018-HU-01

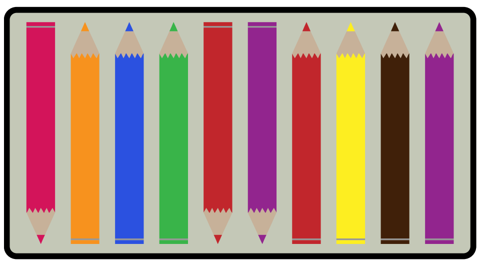
3/4 mittel
5/6 –
7/8 –
9/10 –
11–13 –

Ada hat eine Schachtel mit 10 Farbstiften. Einige zeigen nach oben, einige zeigen nach unten. Ada möchte, dass alle Farbstifte nach oben zeigen.

Als Spiel dreht sie immer nur zwei oder mehr nebeneinanderliegende Farbstifte gleichzeitig um. Im folgenden Beispiel dreht sie den vierten, fünften, sechsten und siebten Farbstift gleichzeitig um.



Ada möchte, dass alle Farbstifte in der Schachtel nach oben zeigen. Sie will so wenig Schritte wie möglich machen. Mit wie wenig Schritten schafft sie es?



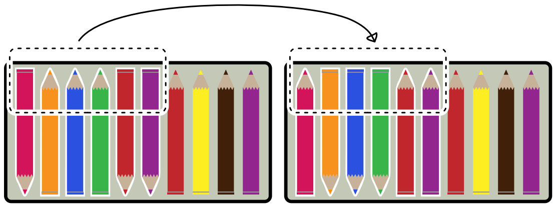
Adas Farbstifte

2018-HU-01

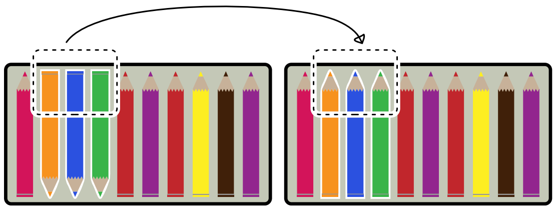
Lösung

Es genügt, zwei mal Farbstifte nach der Spielregel umzudrehen. Einmal würde nicht genügen, da die Farbstifte nicht alle nebeneinander liegen.

Wenn Ada aber die Stifte 1 bis 6 umdreht...



...und danach die Stifte 2 bis 4 umdreht, ...



liegt alles wie gewünscht.

Das ist Informatik!

Ada hat wahrscheinlich genug Zeit, viel mehr als zwei Schritte zu machen, damit die Stifte wie gewünscht liegen. Und wenn sie keine Lust mehr hat, kann sie natürlich auch einen einzelnen Stift umdrehen.

Computer haben es nicht so einfach. Wenn man einen Computer programmiert hat, hält er sich an die Spielregeln. Computer haben es auch nicht mit einigen wenigen Stiften zu tun, sondern müssen vielleicht dieselbe Spielregel auf ganz viele Daten anwenden.

Ein Beispiel, wo das wichtig ist, ist der Festspeicher von Computern. Heutzutage baut man in der Regel eine *SSD (Solid State Disk)* ein. Das Löschen einer einzelnen Speicherinformation geht nicht alleine, sondern nur in Form von Blöcken. Beim Schreiben muss der Computer also darauf achten, dass der ganze Block unbenutzt ist, sonst muss er erst die benutzten Teile lesen, dann den Block löschen und dann den Block mit den alten und neuen Daten neu beschreiben. Das ist viel langsamer, als wenn der Computer unbenutzte Blöcke kennt und schon einmal löscht, während er sonst nichts zu tun hat. Wenn er viel schreiben muss, lohnt es sich für ihn, die Daten immer in ganzen Blöcken zu speichern, denn ob er nun nur eine einzelne Speicherinformation oder einen ganzen Block schreibt, ist für ihn gleich schnell.

Stichwörter: Effizienz

https://www.codechef.com/problems/ADACRA

https://de.wikipedia.org/wiki/Effizienz_(Informatik)

https://de.wikipedia.org/wiki/Landau-Symbole

Karten drehen

2018-BE-01a

Jemand schenkt dir einen Satz gleicher Spielkarten. Die Karten sehen so aus:

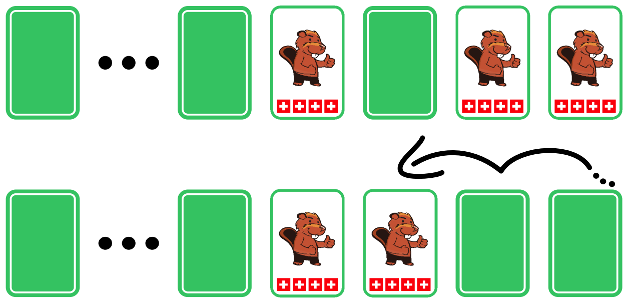
Aufgedeckt: Verdeckt:

Mit diesen Karten kannst du „Drehen“ spielen. Dafür legst du eine Reihe Karten vor dir aus. In einem Spielzug gehst du diese Karten von rechts nach links so durch:

- Ist die aktuelle Karte aufgedeckt, drehe sie um.
- Ist die aktuelle Karte verdeckt, drehe sie um. Damit ist der Spielzug beendet, die übrigen Karten bleiben unverändert.

Ein Spielzug könnte zum Beispiel sein:

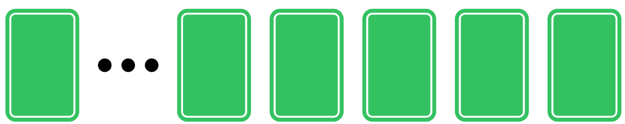
Vorher:



Nachher:

Die beiden rechten verdeckten Karten drehst du um. Die nächste Karte ist verdeckt. Du deckst sie auf und damit ist der Spielzug beendet.

Diesmal beginnst das Spiel mit 16 verdeckten Karten.



Wie viele Karten sind nach 16 Spielzügen aufgedeckt?

Karten drehen

2018-BE-01a

Lösung

Es ist genau eine Karte aufgedeckt.

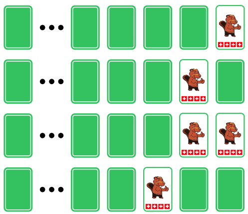
Man kann sich die Reihe von gleichen Karten, die entweder verdeckt oder aufgedeckt sein können, als Binärzahl vorstellen. Binärzahlen bestehen nur aus den Ziffern 0 und 1. Eine verdeckte Karte stellt beispielsweise die Ziffer 0 dar und eine aufgedeckte Karte die Ziffer 1. Analog zum üblichen Zehnersystem gibt jede Stelle einer Binärzahl an, ob die passende Zweierpotenz in den Wert der Zahl einzurechnen ist oder nicht. Ist z. B. die dritte Stelle (von rechts) einer Binärzahl mit einer 1 besetzt, ist die dritte Zweierpotenz zum Wert der Zahl zu addieren – also 2^2, denn $1 = 2^0$.

Binärzahlen werden auf diese Weise um 1 hochgezählt. Man beginnt mit der Ziffer ganz rechts:

- Ist die aktuelle Ziffer eine 0, mache daraus eine 1. Damit ist die gesamte Zahl um 1 hochgezählt.
- Ist die aktuelle Ziffer eine 1, mache daraus eine 0 und gehe zur nächsten Ziffer nach links für den Übertrag.

Das entspricht genau einem Spielzug im Spiel „Drehen". Ein Spielzug erhöht also den Wert der Binärzahl, die durch die Kartenreihe dargestellt wird, um 1. Die verdeckten Karten zu Beginn stellen die Binärzahl dar, die nur aus Nullen besteht, also den Wert 0 hat.

Das folgende Bild zeigt die Ergebnisse der ersten vier Spielzüge, die also den Zahlen von 1 bis 4 entsprechen. Man kann sehen, dass bei den Zahlen 1, 2 und 4 (also den Zweierpotenzen 2^0, 2^1 und 2^2) genau eine Karte aufgedeckt ist. Das heisst, dass bei einer Binärzahl, die eine Zweierpotenz als Wert hat, nur an der Stelle, die dieser Zweierpotenz entspricht, eine 1 ist.



Nach 16 Spielzügen erhalten wir also die Darstellung einer Binärzahl mit dem Wert 16. Da $16 = 2^4$ ist, hat diese Binärzahl an der fünften Stelle von rechts eine 1 und sonst nur Nullen: 0000000000010000. In der Darstellung mit den Karten ist also genau diese eine Karte aufgedeckt:

Das ist Informatik!

Die kleinste Speichereinheit heutiger Computer kann nur zwei Werte unterscheiden: AN oder AUS, WAHR oder FALSCH, 0 oder 1. Alle Daten, die in einem Computer gespeichert und verarbeitet werden, müssen wir als Reihen binärer Ziffern sehen, letztlich also als *Binärzahlen*. Deshalb haben *Operationen* auf Binärzahlen für die Informatik eine grosse Bedeutung.

Seitdem es Computer gibt, wird das Abarbeiten solche Operationen möglichst effizient gebaut. Es gibt Operationen, die zwei Binärzahlen miteinander verknüpfen, wie etwa die Rechenoperationen Addition oder Multiplikation. Es gibt aber auch Operationen, die eine einzelne Binärzahl verändern, etwa das Verschieben aller Ziffern um eine Position nach links oder eben das Hochzählen, also die Addition um 1 wie in dieser Aufgabe.

Gute *Prozessoren* zeichnen sich dadurch aus, dass sie solche Operationen schnell und energiesparend ausführen, und zwar millionenfach pro Sekunde. Es ist dann die Aufgabe des Programmierers und seiner Werkzeuge, komplizierte Abläufe, auch diese einfachen Operationen, zu reduzieren, so dass der Benutzer beliebige Programme verwenden kann.

Stichwörter: Binärzahlen

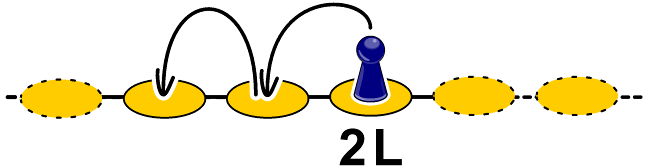
https://de.wikipedia.org/wiki/Dualsystem

https://de.wikipedia.org/wiki/Synchronzähler

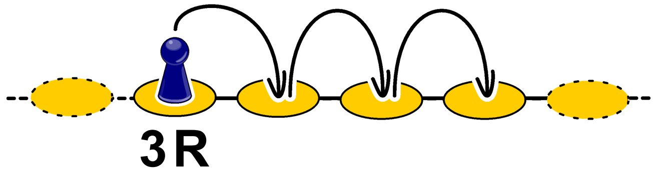
https://de.wikipedia.org/wiki/Asynchronzähler

Wie bei jedem Hüpfspiel muss man auch hier Felder nach bestimmten Regeln abhüpfen. Bei diesem Hüpfspiel gehört zu jedem Feld eine Regel. Es gibt drei Arten von Regeln:

- nL: n Felder nach links hüpfen, 2L bedeutet also, zwei Felder nach links zu hüpfen:

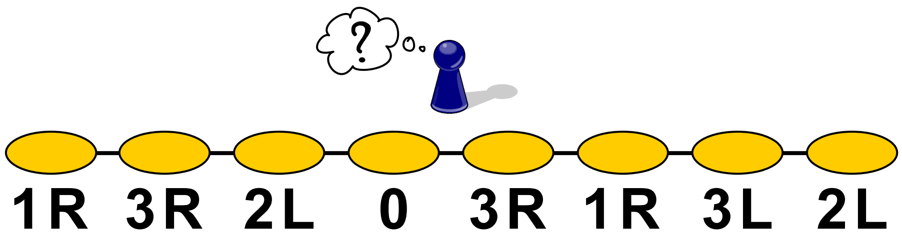


- *n*R: *n* Felder nach rechts hüpfen, 3R bedeutet also, drei Felder nach zu rechts hüpfen:



- 0: nicht mehr weiter hüpfen.

Auf welchem Feld muss man starten, damit man nach dem Spiel auf jedem Feld einmal gewesen ist?



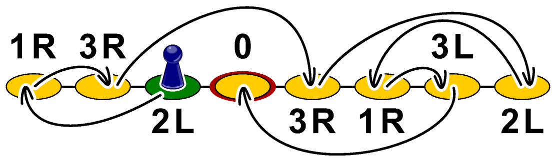
Hüpfspiel
2018-CA-06

3/4 –
5/6 –
7/8 –
9/10 mittel
11–13 einfach

Lösung

Wenn man auf dem dritten Feld von links („2L") startet, wird man am Ende auf jedem Feld einmal gewesen sein.

Man kommt darauf, indem man für das Feld „0" das Feld sucht, von welchem aus man es erreichen kann. Das ist in diesem Fall das zweite Feld von rechts („3L"). Dieses wiederum erreicht man vom dritten Feld von rechts („1R"), dieses wiederum vom Feld ganz rechts („2L"), dieses vom vierten Feld von rechts („3R"), das vom zweiten Feld von links („3R"), dieses nun vom Feld ganz links („1R") und zuletzt bleibt nur noch das dritte Feld von links („2L"), das tatsächlich wie gewünscht das Feld ganz links erreichen lässt.



Dieses Markieren der Hüpfwege mit Pfeilen macht aus den Feldern einen gerichteten Graphen, den man nun lediglich von 0 an rückwärts durchgehen muss, um beim dritten Feld von links als Startfeld zu landen.

Das ist Informatik!

In der Informatik funktioniert das Speicherprinzip *„Verkettete Liste"* ähnlich wie die Hüpffelder: im Arbeitsspeicher werden Objekte mit Verweisen auf die Speicheradresse des nächsten Objekts gespeichert. So kann die Speicherverwaltung ein einzelnes Objekt beliebig im Arbeitsspeicher hinterlegen und muss nicht zeitaufwendig für alle Objekte einen zusammenhängenden Platz im Arbeitsspeicher erstellen. Der Programmierer wiederum muss sich um nichts kümmern, ausser dass er bei der Speicherverwaltung einen Speicher in der entsprechenden Grösse reserviert.

Was passiert aber, wenn Objekte im Speicher nicht mehr gebraucht werden? Während sich früher die Programmierer darum kümmern mussten, dass der Speicher wieder freigegeben wird (was leider häufig schief ging, so dass Programme mit der Zeit immer mehr Speicher verschwendeten und dann irgendwann aus Speichermangel abstürzten), haben moderne Programmiersprachen hierfür eine Art Müllabfuhr, die *„Garbage Collection"*, die regelmässig überprüft, ob Objekte im Speicher noch von anderen Objekten referenziert werden (ob es also noch Verweise auf sie gibt). Da manchmal grosse Strukturen von Objekten nicht mehr referenziert werden, muss genau wie im Beispiel zurückverfolgt werden, von welchem „Startfeld" aus auf die anderen Objekte verwiesen wird.

Stichwörter: Schlange, Speicherverwaltung, GOTO

https://de.wikipedia.org/wiki/Garbage_Collection

Brückenbau

2016-FR-03

3/4 –

5/6 –

7/8 mittel

9/10 einfach

11–13 -

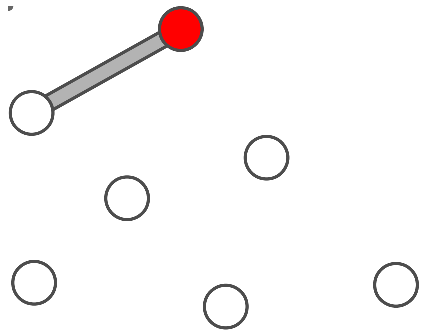
Der Biber-Opa ist ein wenig wasserscheu geworden. Er möchte von seiner Burg zu allen anderen Burgen der Biber-Familie über Brücken gehen können. Die Biber meinen es gut mit Opa und wollen beim Brückenbauen folgendes beachten:

- Opa soll von seiner Burg aus höchstens über zwei Brücken gehen müssen.
- Neben der Brücke, mit der man zu einer Burg kommt, dürfen höchstens zwei weitere Brücken davon wegführen.

Die Biber beginnen mit einem Brückenplan. Sie zeichnen alle Burgen als Kreise. Opas Burg ist ein rot ausgefüllter Kreis. Eine erste Brücke von Opas Burg aus zeichnen sie ein. Aber dann wissen sie nicht mehr weiter.

Vervollständige den Plan so, dass er alle Bedingungen erfüllt.

Dazu gibt es mehrere Möglichkeiten. Auf jeden Fall werden fünf weitere Brücken benötigt.



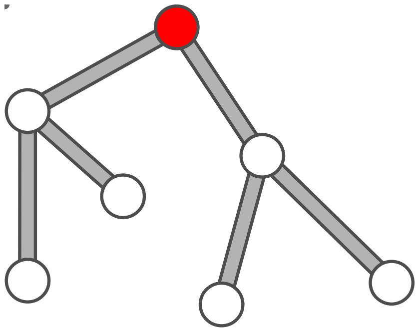
Brückenbau

2016-FR-03

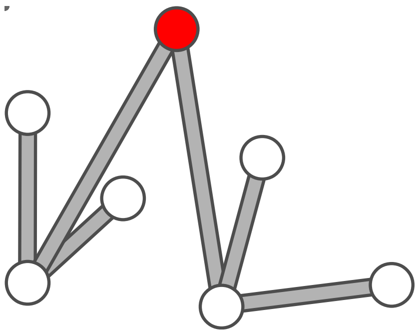
Lösung

Der folgende Plan erfüllt alle Bedingungen, denn:

- vom roten Kreis aus ist jeder andere Kreis über höchstens zwei Linien zu erreichen
- von jedem Kreis gehen höchstens drei Linien aus.



Es gibt auch noch andere Pläne, die richtig sind, zum Beispiel:



Das ist Informatik!

Obwohl es viele mögliche Lösungen gibt, haben sie alle denselben Aufbau. Von der roten Burg aus gehen zwei Brücken zu zwei anderen Burgen und von diesen beiden anderen Burgen aus gehen je zwei Brücken zu den verbleibenden vier Burgen. Gäbe es eine Burg mehr, könnte man diese nicht unter Einhalten der Regeln auch noch erreichen.

Die Biber bauen hier einen sogenannten *Baum* auf: alle *Knoten* (die Burgen) können erreicht werden, indem man *Kanten* (Brücken) entlang läuft. Die beiden Regeln stellen sicher, dass ein besonderer Baum erzeugt wird: die Tatsache, dass nur zwei Kanten weiterführen dürfen, erzeugen einen *Binärbaum*. Die Tatsache, dass man maximal zwei Kanten bis zu jedem Knoten entlang laufen darf, stellt sicher, dass der Baum *minimal* ist.

Stichwörter: Binärbaum, Graphentheorie

https://de.wikipedia.org/wiki/Baum_(Graphentheorie)

https://de.wikipedia.org/wiki/Binärbaum