

Informatik- Biber

AUFGABEN 2017

Der Wettbewerb zum digitalen Denken.



www.bwinf.de



bwinf.de/biber

Herausgeber Wolfgang Pohl, BWINF; Hans-Werner Hein, BWINF

Der Aufgabenausschuss Informatik-Biber 2017

Hans-Werner Hein, BWINF Bonn
Ulrich Kiesmüller, Simon-Marius-Gymnasium Gunzenhausen
Wolfgang Pohl, BWINF Bonn
Kirsten Schlüter, St.-Emmeram-Realschule Aschheim
Michael Weigend, Holzkamp-Gesamtschule Witten

Die deutschsprachigen Fassungen der Aufgaben wurden auch in Österreich und der Schweiz verwendet. An ihrer Erstellung haben mitgewirkt:

Andrea Adamoli, Università della Svizzera italiana
Lorenz Baues, BWINF Bonn
Wilfried Baumann, Österreichische Computer Gesellschaft
Robert Czechowski, BWINF Bonn
Christian Datzko, Wirtschaftsgymnasium und Wirtschaftsmittelschule Basel
Susanne Datzko, freischaffende Graphikerin
Olivier Ens, Freis Schulen
Hanspeter Erni, Pädagogische Hochschule Luzern
Gerald Futschek, Technische Universität Wien
Peter Garscha, Technische Universität Wien
Martin Guggisberg, Pädagogische Hochschule FHNW
Urs Hauser, ETH Zürich
Juraj Hromkovic, ETH Zürich
Ivana Kosirová, ETH Zürich
Bernd Kurzmann, BG / BRG / WISKU 11, Wien
Regula Lacher, ETH Zürich
Katharina Resch-Schobel, Österreichische Computer Gesellschaft
Martin Stangl, Student Pädagogische Hochschule FHNW

Der Informatik-Biber

ist ein Projekt der Bundesweiten Informatikwettbewerbe (BWINF).
BWINF ist eine Initiative der Gesellschaft für Informatik (GI),
des Fraunhofer-Verbunds IUK-Technologie und
des Max-Planck-Instituts für Informatik.
BWINF wird vom Bundesministerium für Bildung und Forschung (BMBF)
gefördert. Die Bundesweiten Informatikwettbewerbe gehören zu den
von den Kultusministerien empfohlenen Schülerwettbewerben und stehen
unter der Schirmherrschaft des Bundespräsidenten.

Einleitung

Der Informatik-Biber ist ein Online-Test mit Aufgaben zur Informatik. Er erfordert Köpfchen, aber keine Vorkenntnisse.

Der Informatik-Biber will das allgemeine Interesse für das Fach Informatik wecken und gleichzeitig die Motivation für eine Teilnahme an Informatikwettbewerben stärken. Schülerinnen und Schüler, die mehr wollen, sind herzlich eingeladen, sich anschließend am Jugendwettbewerb Informatik und auch am Bundeswettbewerb Informatik zu versuchen (siehe Seite 5).

Der Informatik-Biber findet jährlich im November statt. An der 11. Austragung im Jahr 2017 beteiligten sich 1.898 Schulen mit 341.241 Schülerinnen und Schülern. Die Möglichkeit, auch in Zweierteams zu arbeiten, wurde gern genutzt.

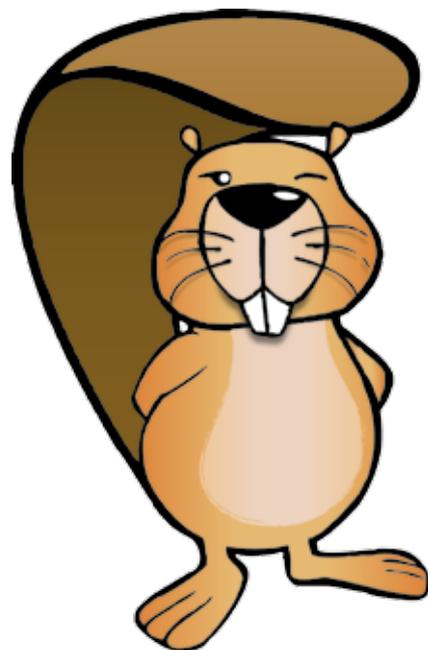
Die Online-Teilnahme am Informatik-Biber 2017 war mit Desktops, Laptops und Tablets möglich. Etwa die Hälfte der Antworteingaben waren multiple-choice. Verschiedene andere Interaktionsformen machten die Bearbeitung abwechslungsreich. In diesem Biberheft ist die Dynamik der Aufgabenbearbeitung nicht vorführbar. Darum geben Handlungstipps in den Aufgabenstellungen und Bilder von Lösungssituationen davon eine Vorstellung. Der Umgang mit dem Wettbewerbssystem selbst konnte in den Wochen vor der Austragung online geübt werden.

Der Informatik-Biber 2017 wurde in fünf Altersgruppen durchgeführt. In den Klassenstufen 3 bis 4 waren innerhalb von 30 Minuten 9 Aufgaben zu lösen, jeweils drei in den Schwierigkeitsstufen leicht, mittel und schwer. In den Klassenstufen 5 bis 6, 7 bis 8, 9 bis 10 und 11 bis 13 waren innerhalb von 40 Minuten 15 Aufgaben zu lösen, jeweils fünf in den Schwierigkeitsstufen leicht, mittel und schwer.

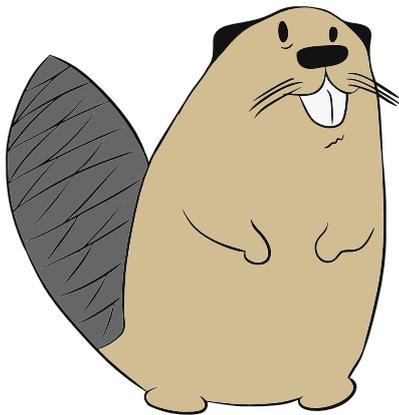
Die 39 Aufgaben des Informatik-Biber 2017 sind auf Seite 6 gelistet, nach ungefähr steigender Schwierigkeit und mit einer informatischen Klassifikation ihres Aufgabenthemas. Ab Seite 7 folgen die Aufgaben nach ihrem Titel alphabetisch sortiert. Im Kopf sind die zugeordneten Altersgruppen und Schwierigkeitsgrade vermerkt. Eine kleine Flagge gibt an, aus welchem Bebras-Land die Idee zu dieser Aufgabe stammt. Der Kasten am Aufgabenende enthält Erläuterungen zu den Lösungen und Lösungswegen sowie eine kurze Darstellung des Aufgabenthemas hinsichtlich seiner Relevanz in der Informatik.

Die Veranstalter bedanken sich bei allen Lehrkräften, die mit großem Engagement ihren Klassen und Kursen ermöglicht haben, den Informatik-Biber zu erleben.

Wir laden die Schülerinnen und Schüler ein, auch 2018 wieder beim Informatik-Biber mitzumachen. Der genaue Termin und vieles mehr wird über die Website bwinf.de und per E-Mail an die Koordinatorinnen und Koordinatoren bekannt gegeben.



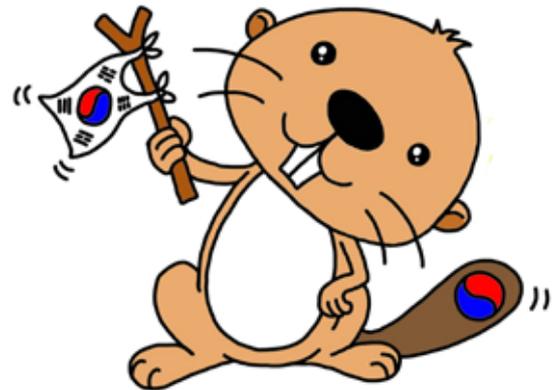
Bebras: International Challenge on Informatics and Computational Thinking



Der ungarische Biber

Der deutsche Informatik-Biber ist Partner der internationalen Initiative Bebras. 2004 fand in Litauen der erste Bebras Challenge statt. 2006 traten Estland, die Niederlande und Polen der Initiative bei, und auch Deutschland veranstaltete im Jahr der Informatik als „El: Spiel blitz!“ einen ersten Biber-Testlauf. Seitdem kamen viele Bebras-Länder hinzu. Zum Drucktermin sind es weltweit 56, und weitere Länderteilnahmen sind in Planung. Insgesamt hatte der Bebras Challenge 2017 international weit über zwei Millionen Teilnehmerinnen und Teilnehmer.

Die Bebras-Community erarbeitet jedes Jahr auf einem internationalen Workshop anhand von Vorschlägen der Länder eine größere Auswahl möglicher Aufgabenideen. Die 39 Aufgabenideen des Informatik-Biber 2017 stammen aus 20 Ländern: Aserbaidschan, Deutschland, Großbritannien, Iran, Italien, Japan, Kanada, Korea, Kroatien, Niederlande, Österreich, Polen, Rumänien, Russland, Schweiz, Serbien, Slowakei, Slowenien, Tschechien und Ungarn.



Der koreanische Biber

Deutschland nutzt zusammen mit einer Vielzahl anderer Länder zur Durchführung des Informatik-Biber ein gemeinsames Online-System. Dieses „International Bebras Challenge System“ wird von der niederländischen Firma Eljakim IT betrieben und fortentwickelt.

Informationen über die Aktivitäten aller Bebras-Länder finden sich auf der Website bebras.org.



Die schwedischen Biber



Bundesweite Informatikwettbewerbe



Bundesweite
Informatikwettbewerbe

Bei jungen Menschen das Interesse für Informatik wecken, Begabungen entdecken und fördern: das ist das Ziel der Bundesweiten Informatikwettbewerbe (BWINF), an denen im Jahr 2017 über 350.000 junge Menschen teilnahmen. Der Informatik-Biber ist das BWINF-Einstiegsformat; außerdem werden noch drei weitere Wettbewerbe angeboten:

 Informatik-Biber

 Jugendwettbewerb
Informatik

 Bundeswettbewerb
Informatik

 Informatik-Olympiade

Jugendwettbewerb Informatik

Der Jugendwettbewerb Informatik (JwInf) wurde 2017 zum ersten Mal ausgerichtet. Er richtet sich an Kinder und Jugendliche, die erste Programmiererfahrungen sammeln und vertiefen möchten. Er ist in den ersten Runden ein reiner Online-Wettbewerb, genauso wie der Informatik-Biber. Empfohlen wird eine Teilnahme ab der Jahrgangsstufe 5; die dafür nötigen Kenntnisse können auf der JwInf-Plattform erworben werden (wettbewerb.jwinf.de).

Bundeswettbewerb Informatik

Der Bundeswettbewerb Informatik (BwInf) wurde 1980 von der Gesellschaft für Informatik e.V. (GI) auf Initiative von Prof. Dr. Volker Claus ins Leben gerufen. Dieser traditionsreichste BWINF-Wettbewerb beginnt jedes Jahr im September. Die Aufgaben der ersten und zweiten Runde werden zu Hause selbstständig bearbeitet, einzeln oder in einer Gruppe. In der zweiten Runde ist dann eigenständiges Arbeiten gefordert. Die ca. dreißig bundesweit Besten werden zur dritten Runde, einem Kolloquium, eingeladen. Allen Teilnehmenden stehen weitergehende Fördermaßnahmen offen. Die Siegerinnen und Sieger werden ohne weiteres Verfahren in die Studienstiftung des deutschen Volkes aufgenommen.

Internationale Informatik-Olympiade

Die Jüngeren unter den BwInf-Finalisten und einige ausgewählte Teilnehmende der zweiten Runde können sich in mehreren Trainingsrunden sowie bei Vorbereitungswettbewerben im europäischen Ausland für das vierköpfige deutsche Team qualifizieren, das im Folgejahr an der Internationalen Informatik-Olympiade (IOI) teilnimmt.

Austausch

Die Teilnahme an BWINF-Wettbewerben eröffnet Möglichkeiten zum Austausch mit Gleichgesinnten. Erste Anknüpfungspunkte bieten „BWINF – Informatik erleben“ bei Facebook, die BWINF-Accounts bei Twitter und Instagram, das Informatik-Jugendportal Einstieg Informatik mit seiner Community und die BWINF-Website. Die mehr als 30 Jahrgänge von BwInf-Teilnehmenden bilden ein wachsendes Netzwerk, vor allem im BwInf Alumni und Freunde e.V. Nach der ersten BwInf-Runde lernen sich viele Teilnehmende bei Informatik-Workshops von Hochschulen und Unternehmen kennen.

Träger und Förderer

BWINF ist eine Initiative der Gesellschaft für Informatik (GI), des Fraunhofer-Verbunds IUK-Technologie und des Max-Planck-Instituts für Informatik. BWINF wird vom Bundesministerium für Bildung und Forschung (BMBF) gefördert. Die Bundesweiten Informatikwettbewerbe gehören zu den von den Kultusministerien empfohlenen Schülerwettbewerben und stehen unter der Schirmherrschaft des Bundespräsidenten.

Aufgabenliste

Dies sind die 39 Aufgaben des Informatik-Biber 2017, geordnet nach ungefähr steigender Schwierigkeit und gelistet mit einer Klassifikation ihres informatischen Inhalts.

Titel	Thema	Seite
Vogelhaus	Objektstrukturen, Objekteigenschaften	59
Nachrichtendienst	Internet-Protokoll, Datenpakete	43
Ab durch die Lücke!	Programmieren, Sequenz	7
Fünf Hölzchen	Programmieren, Anweisung, Zustandsänderung	19
Lichtkunst	Programmieren, Speicher, Reihenfolge	40
Ehrenname	Zeichenersetzung, Codierung, Verschlüsselung	17
Parkplätze	Logik, Funktion ODER	46
Lecker!	Programmieren, Speicherinhalte tauschen	38
Klammerschmuck	Syntax, Klammern	29
Platz! Tausch!	Algorithmik, Sortieren durch Platztausch	49
Bitte lächeln!	Mustererkennung, künstliche Intelligenz, Verlässlichkeit	15
Mauern öffnen	Algorithmik, Breitensuche	41
Kreiselstadt	Programmieren, homogene Struktur	32
Kreisch!	Parallelität, Prozess, Ressourcen	30
Saftladen	Algorithmik, Optimierung, Matching	52
Stock und Schild	Algorithmik, Backtracking, logisches Denken	56
Honomakato	Datenstruktur, zusammenhängender Graph, Brücke	25
Gartentor	Binärcode, Zeichensysteme	21
Schülerzeitung	Datenstruktur, Tabelle	54
Geheime Bestellung	Kryptografie, Cäsar-Chiffre, Vigenère	23
Belastungstest	Datenanalyse, Balkendiagramm	12
Kugelrampe	Datenstruktur, Stapelspeicher (LIFO)	33
Wortabstand	Algorithmik, Codierung, Abstandsmaß für Zeichenketten	60
BikeFun	Formale Sprachen, Klammern	13
Pumpsystem	Fehler, Testen	50
Alarm im Museum!	Datenanalyse, Zustände, Objektschutz	8
Kurzes Programm	Programmieren, Wiederholung, bedingte Anweisung	35
Der neue Song	Graph, soziales Netzwerk	16
Ersetzungen	Programmieren, Speicherinhalte tauschen	18
Pizzeria Biberia	Algorithmik, Planung, Greedy	47
Arabot	logisches Denken, Algorithmik	10
Ladezeit	Datenübertragung, Netzneutralität	37
Opas Marmelade	Planen, parallele Prozesse	44
Punkte sammeln	Algorithmik, Skalierbarkeit, dynamische Programmierung	51
Viertel	verlustfreie Datenkompression, Quadtree	57
Zerteile den Code	Codierung, Präfixcode	61
Kürzeste Wege	logisches Denken, Deduktion, unscharfes Schließen	36
Hunger!	Algorithmik, Planung, Ford-Fulkerson	27
Ziffernsegmente	Algorithmik, Teilprobleme, Redundanz	62



Ab durch die Lücke!

Der dreieckige Roboter soll vom roten Feld zum grünen Zielfeld gehen.

Programmiere den Roboter!

Ziehe dazu Programmbausteine nach rechts und füge sie in der richtigen Reihenfolge zusammen.

Das bedeuten die Knöpfe:



... Programm testen



... Programm testen und speichern



... Programm anhalten/zurücksetzen



So ist es richtig:

Damit der Roboter von seinem Startfeld auf das Zielfeld gelangt, muss er zunächst ein Feld nach vorne gehen, damit er neben der Lücke in der Mauer steht. Wenn er sich nach links dreht und zwei Felder nach vorne geht, ist er durch die Lücke in der Mauer durch. Nun braucht er sich nur noch nach rechts zu drehen und drei Felder nach vorne zu gehen, um beim Zielfeld anzukommen.

Das ist Informatik!

In der mobilen Robotik ist die Navigation ein allgemeines Problem. Labyrinth-Lösungen kommen zwar nicht häufig vor, aber erfordern ähnliche Computational-Thinking-Fähigkeiten. Um dieses Problem zu lösen wird ein automatisierter Roboter verwendet. Labyrinth können von unterschiedlicher Art sein: Mit oder ohne Schleifen, mit oder ohne Rastersystemen. Der Roboter dieser Aufgabe hat keine Sensoren, um beispielsweise Mauern zu erkennen, er ist quasi ‚blind‘. Daher muss er mit ganz präzisen Anweisungen versorgt werden, damit er die Lücke und somit den korrekten Weg findet.



Alarm im Museum!

Im Museum für moderne Holzplastiken gibt es ein intelligentes Sicherheitssystem. Das System ermittelt jede Minute, wie viele Personen in jedem Raum sind, und trägt die Zahlen in eine Tabelle ein. Außerdem prüft es anhand der Zahlen, ob ein Eindringling im Museum ist. Ein Eindringling ist eine Person, die in das Museum gelangt ist, aber nicht die Eingangstür benutzt hat.

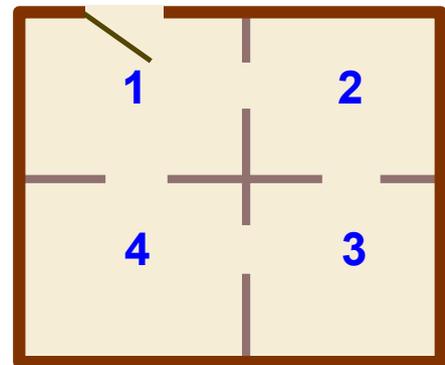
Sobald das System meint, dass ein Eindringling im Museum ist, schlägt es Alarm.

Links siehst du die Tabelle des Systems für die Minuten 10:01 bis 10:07.

Rechts siehst du den Raumplan des Museums mit den Nummern der Räume.

Das Museum ist groß: Von einer Minute zur nächsten kann man es höchstens durch eine Tür schaffen.

Zeit	Raum 1	Raum 2	Raum 3	Raum 4
10:01	2	0	0	0
10:02	3	0	0	0
10:03	2	1	0	0
10:04	4	1	1	0
10:05	2	2	3	0
10:06	5	2	2	1
10:07	4	1	2	2



In einer der Minuten in der Tabelle schlägt das System Alarm. In welcher?

10:01

10:02

10:03

10:04

10:05

10:06

10:07

**10:05 ist die richtige Antwort:**

Das System schlägt um 10:05 Alarm. Zu diesem Zeitpunkt waren in Raum 3 zwei Personen mehr als in der Minute davor. In den Nachbarräumen war dann (also um 10:04) aber nur eine Person, und zwar in Raum 2. Folglich muss jemand von außen in Raum 3 eingedrungen sein, ohne den Eingang des Museums zu benutzen.

Zu den anderen Zeitpunkten wird kein Eindringling erkannt:

Um 10:01 sind 2 Personen in Raum 1; sie können den Eingang des Museums benutzt haben.

Von 10:01 bis 10:02 kann 1 Person durch den Eingang des Museums gekommen sein.

Von 10:02 bis 10:03 kann 1 Person von Raum 1 nach Raum 2 gegangen sein.

Von 10:03 bis 10:04 können 1 Person von Raum 2 nach Raum 3 und 1 Person von Raum 1 nach Raum 2 gegangen sowie 3 Personen durch den Eingang gekommen sein.

Von 10:05 bis 10:06 können 1 Person von Raum 3 nach Raum 4 gegangen und 3 Personen durch den Eingang gekommen sein.

Von 10:06 bis 10:07 können 1 Person von Raum 1 nach Raum 4 und 1 Person von Raum 2 nach Raum 1 gegangen sein sowie 1 Person das Museum durch den Eingang verlassen haben.

Das ist Informatik!

Informatiksysteme, die die Anzahl von Personen in kritischen Bereichen überwachen, findet man zum Beispiel auf Flughäfen. Computerprogramme werten die Live-Bilder von Überwachungskameras aus, erkennen Personen, zählen sie und werten die Zahlen und deren Veränderungen aus. Diese Programme setzen künstliche Intelligenz ein, zum Beispiel um Personen von Gegenständen oder anderen Lebewesen zu unterscheiden. Aber sie verwenden auch einfache Regeln wie in dieser Aufgabe, um Sicherheitsprobleme aufzuspüren.

Wer Personen zählen und die Veränderungen dieser Zahlen auswerten kann, kann dieses Wissen vielfältig nutzen. Bei Veranstaltungen mit sehr vielen Besuchern kann vorhergesagt werden, wo Personenmengen gefährlich groß werden könnten. Im Supermarkt möchte man teure Waren entlang der größten Personenströme platzieren, damit sie gut verkauft werden. In Kombination mit einer gut funktionierenden Gesichtserkennung kann man aber auch versuchen, Personen zu identifizieren und ihre Bewegungen zu verfolgen.



Arabot

Arabot ist ein neuer Spiel-Roboter.

Er fährt auf einem Plan aus schwarzen Strecken und Kreuzungen.



Ein Plan enthält zudem mindestens einen Startpunkt  und eine Ladestation. 

Jede Strecke muss markiert werden. Die Markierung sagt Arabot, welchen Weg er an der nächsten Kreuzung nehmen soll. Es gibt zwei verschiedene Markierungen:



Nimm den am weitesten rechts liegenden Weg!



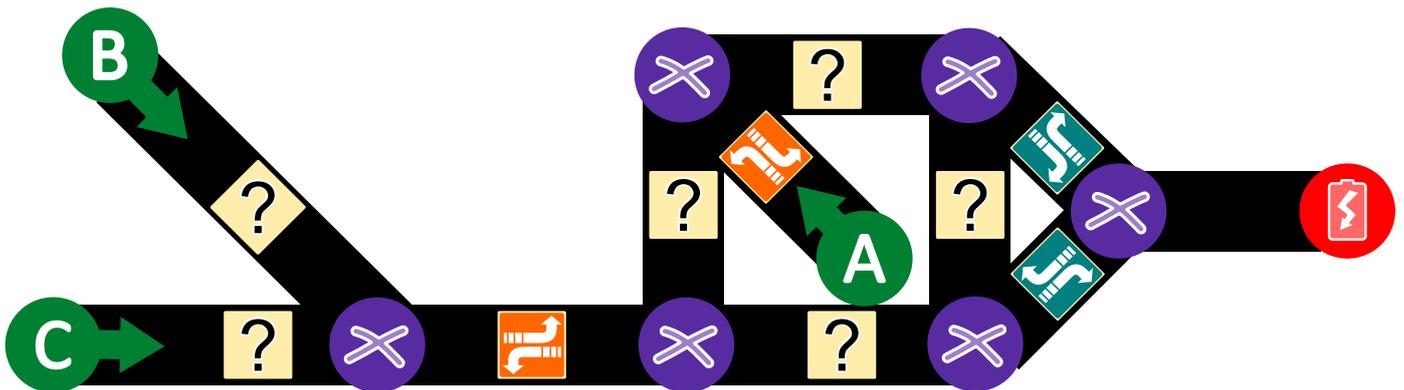
Nimm den am weitesten links liegenden Weg!

Die Strecken müssen so markiert werden, dass Arabot immer die Ladestation erreicht, egal an welchem Startpunkt er beginnt. Wenn Arabot auf einem Startpunkt landet, schaltet er sich ab.

Janine hat für ihren neuen Arabot einen Plan erstellt mit drei Startpunkten A, B und C. Einige Strecken hat sie bereits markiert.

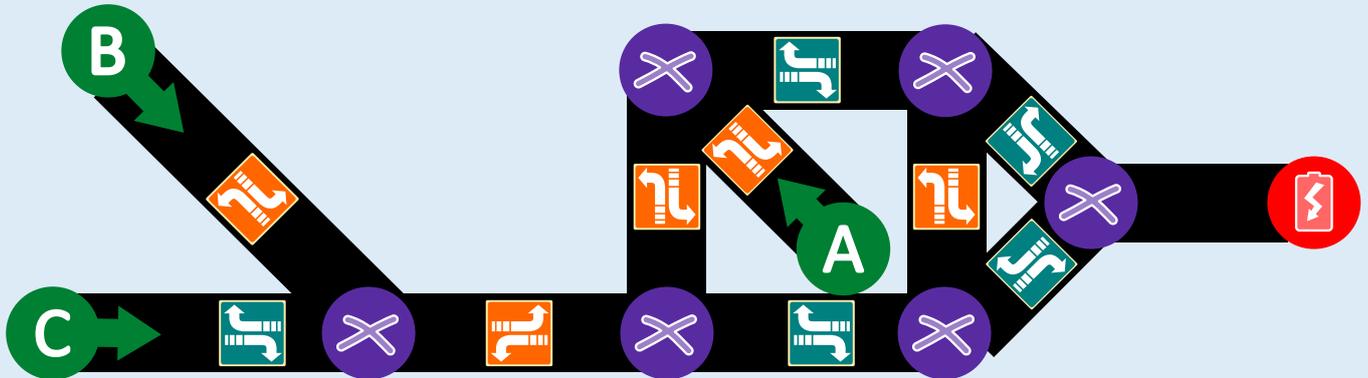
Markiere auch alle weiteren Strecken!

Klicke dazu auf die Stellen mit den Fragezeichen, bis die jeweils benötigte Markierung erscheint.





So ist es richtig:



An der Kreuzung ganz links muss der Arabot weiter nach rechts fahren. Dazu muss die Strecke von Startpunkt B aus mit  und die von C aus mit  markiert werden.

Im Dreieck ganz rechts kann der Arabot nur dann die Ladestation erreichen, wenn er von links über die untere Strecke fährt. Die rechte Strecke des Quadrats in der Mitte muss also mit  markiert werden, die untere Strecke mit .

Die obere Strecke des Quadrats muss mit  und die linke Strecke mit  markiert werden, damit Arabot nicht auf Startpunkt A landen kann.

Die Strecken sind nun so markiert, dass der Arabot von allen Startpunkten aus die Ladestation erreicht.

Das ist Informatik!

In dieser Biberaufgabe geht es zunächst darum, die Angaben zum gewünschten Ziel und die einzuhaltenen Regeln und Anweisungen genau zu verstehen, so dass man sie in allen Situationen korrekt umsetzen kann. Was bedeutet etwa „Nimm den am weitesten rechts liegenden Weg!“, wenn es nur geradeaus und nach links geht? Im zweiten Schritt muss man Einstellungen vornehmen und dann kontrollieren, ob dadurch die geforderten Bedingungen erfüllt sind. Dazu benötigt man entweder eine gute Vorstellungsgabe oder eine alternative Methode der Überprüfung – zum Beispiel mit Papier und Bleistift.

In der Informatik ist es häufig nötig, einen komplizierten Vorgang geistig und mit passenden Hilfsmitteln nachzuvollziehen, den man nicht direkt beobachten kann. Zum Beispiel, wenn man ein Computerprogramm analysiert oder einen Sachverhalt, der später in einem Programm modelliert werden soll. Deshalb ist es für Informatikerinnen und Informatiker wichtig, präzise nachdenken, komplexe Sachverhalte nachvollziehen und beschreiben, überraschende Lösungswege finden und auch unwahrscheinliche Situationen gedanklich berücksichtigen zu können. Wenn du diese Biberaufgabe richtig gelöst hast, spricht einiges dafür, dass auch du das kannst.

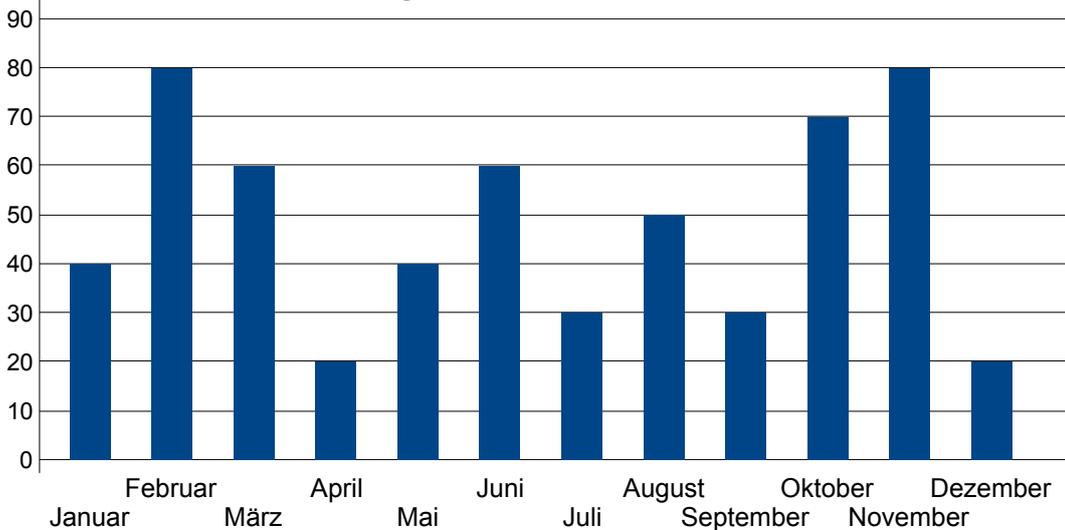


Belastungstest

Für einen kleinen Fluss wird ein neuer Damm geplant. Die Ingenieure studieren die Niederschläge (Regen, Schnee, Hagel) der vergangenen 20 Jahre, um den besten Baumonat zu finden. Sie sind sich über die folgenden Bedingungen einig:

- Die maximale Belastung des Damms muss getestet werden, bevor der Damm freigegeben wird.
- Der Damm muss ein oder zwei Monate vor dem Test gebaut werden.
- Der Baumonat sollte möglichst trocken sein.

Mittlerer Niederschlag der Jahre 1996 bis 2016 in mm



Welchen Monat wählen die Ingenieure?

Januar

April

September

Dezember

„Dezember“ ist die richtige Antwort:

Zuerst werden die Monate mit dem meisten Niederschlag bestimmt. Das sind Februar und November. Für diese beiden Monate werden jeweils die beiden Vormonate betrachtet. Vor dem Februar liegen Dezember und Januar. Vor dem November liegen September und Oktober. Der trockenste dieser Monate ist der Dezember.

Das ist Informatik!

In dieser Biberaufgabe wird in einer Datenmenge ein Datensatz gesucht, der bezüglich einer Reihe von Bedingungen optimal ist. Dabei besteht die Datenmenge aus 12 Paaren der Form (<Monat>, <Niederschlagsmenge>). Weil die Datenmenge klein und in einem Diagramm übersichtlich dargestellt ist und nur wenige Bedingungen erfüllt werden müssen, lässt sich der optimale Datensatz ohne besondere Hilfsmittel schnell bestimmen.

Es gibt viele Beispiele für derartige Suchen nach optimalen Datensätzen, etwa: Welche Tankstelle mit Backshop in einer Entfernung von bis zu 5 km, die auch nachts geöffnet ist, hat den niedrigsten Benzinspreis? Welcher Online-Shop bietet eine Action-Kamera mit Garantie und kostenfreier Rücksendung für den günstigsten Preis an? Wo ist der beste Standort für zehn Windräder in einer bestimmten Region? In all diesen Fällen müssen sehr viel größere Datenmengen durchsucht werden. Die Datensätze selbst sind deutlich komplizierter, und es sind mehr Bedingungen zu überprüfen. Wie gut, dass die Informatik viele Verfahren zur Suche nach optimalen Datensätzen kennt und dafür sorgen kann, dass sie mit Hilfe von Computern auch auf sehr große Datenmengen angewendet werden können.



C)



D)

**Antwort C ist richtig:**

C ist der einzige nach den Regeln befahrbare Parcours.

Um Parcours C zu befahren, kann man auf den flachen Abschnitten die Geschwindigkeit so verändern: + – – (also erhöhen, verringern und wieder verringern). Damit beträgt die Geschwindigkeit am Ziel 0 km/h. Das gilt auch, wenn die Geschwindigkeit so verändert wird: – + – oder – – +.

Die anderen Parcours sind nicht nach den Regeln befahrbar:

Parcours A: Selbst wenn du auf dem flachen Abschnitt die Geschwindigkeit erhöhst, ist die Fahrt nach vier Abschnitten zu Ende.

Parcours B: Selbst wenn du auf beiden flachen Abschnitten die Geschwindigkeit verringerst, beträgt sie am Ziel mehr als 0 km/h.

Parcours D: Auf dem letzten Abschnitt wird die Geschwindigkeit automatisch um 10 km/h erhöht. Um am Ziel 0 km/h zu betragen, müsste die Geschwindigkeit zu Beginn des letzten Abschnitts negativ sein; aber das ist unmöglich.

Das ist Informatik!

Klammern – ob rund, eckig oder geschweift – kennt man aus der Mathematik. In Ausdrücken wie $[n(n-1)]/2$ oder $(a+b)(a-b)$ beschreiben Klammern, in welcher Reihenfolge die Teilausdrücke auszuwerten sind. Dabei treten Klammern immer paarweise auf, mit einer öffnenden und einer schließenden Klammer. Klammersausdrücke müssen wohlgeformt sein: Jede öffnende Klammer muss eindeutig genau einer darauf folgenden schließenden Klammer und jede schließende Klammer nur einer öffnenden Klammer zugeordnet sein.

So wie in der Mathematik gibt es auch in der Informatik Notationen (bzw. „Sprachen“, wie es in der Informatik heißt) mit Klammern. Ein Beispiel ist HTML, die Beschreibungssprache für Webseiten. In HTML wird z.B. ein Textabsatz von den Markierungen `<p>` und `</p>` umschlossen, und alle HTML-Markierungen selbst haben ein Klammerpaar `<` und `>`. Klammerpaare sind in der Informatik sehr beliebt – und das hat seinen Grund: Sie können von Computern leicht verarbeitet werden, und zwar mit Hilfe der einfachen und effizienten Datenstruktur „Stapel“.

Die Abschnitte in den BikeFun-Parcours sind wie Klammern: Ein Bergab-Abschnitt ist eine öffnende Klammer, und ein Bergauf-Abschnitt ist eine schließende Klammer. Ein flacher Abschnitt ist ein Platzhalter, der durch eine öffnende oder schließende Klammer ersetzt werden kann. Ein BikeFun-Parcours ist befahrbar genau dann, wenn es mindestens eine Ersetzung gibt, durch die ein wohlgeformter Klammersausdruck entsteht. Parcours C kann also auch so beschrieben werden: $(?())$. Er ist befahrbar, weil daraus sogar mehrere wohlgeformte Klammersausdrücke entstehen können: $((()))$ oder $()(())$ oder $()()()$. In der Theoretischen Informatik wird eine Sprache, die nur wohlgeformte Klammersausdrücke enthält, als „Dyck-Sprache“ bezeichnet.



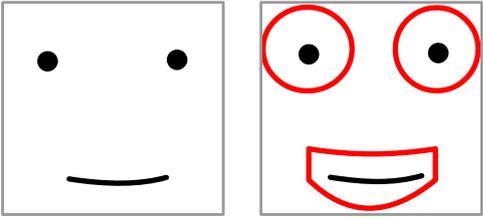
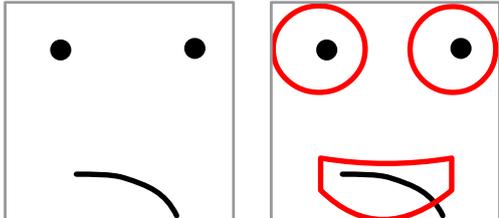
Bitte lächeln!

Eine App soll entscheiden, ob ein gezeichnetes Gesicht lächelt.

Sie benutzt dazu ein „Lächelmuster“ mit Bereichen für Augen und Mund.

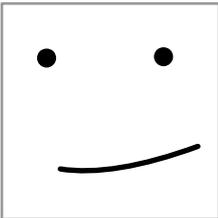
Ein Gesicht lächelt, wenn Augen und Mund ins Lächelmuster passen:

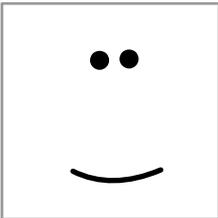
Augen und Mund müssen komplett in den Bereichen des Musters liegen.

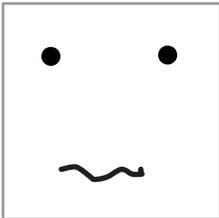
 <p>Lächeln Augen und Mund passen ins Lächelmuster.</p>	 <p>Kein Lächeln Der Mund passt nicht ins Lächelmuster.</p>
---	--

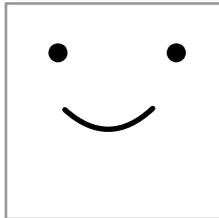
Hier sind vier Gesichter.

Bei welchem Gesicht entscheidet die App, dass es lächelt?

A) 

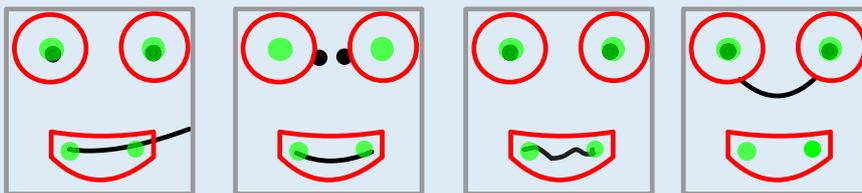
B) 

C) 

D) 

Antwort C ist richtig:

Hier sieht man, wie die vier gezeichneten Gesichter mit dem Lächelmuster geprüft werden. Nur bei Gesicht C passen Augen und Mund ins Lächelmuster.



Das ist Informatik!

Die Informatik entwickelt Softwaresysteme, die in den Bildern einer Videokamera Objekte identifizieren können. Zum Beispiel das Nummernschild eines Autos oder das Gesicht eines Menschen. Das wird mal eine mächtige Technik, bei der auch Methoden der Künstlichen Intelligenz (z.B. deep learning) eingesetzt werden. Mit der Anwendung simpler Regeln wird schon experimentiert.

Diese Biberaufgabe zeigt eine der Schwierigkeiten, die ein Bilderkennungssystem immer haben wird: Es gibt keine klare Grenze zwischen Lächeln und Nichtlächeln. Die Regeln sind hier noch viel zu grob. Das Gesicht, das als lächelnd eingestuft wird, lächelt nicht. Solche Fehler passieren auch in Wirklichkeit bei Systemen zur Gesichtserkennung, die zum Beispiel bei automatischen Passkontrollen oder in Behörden eingesetzt werden. 2016 wurde der Pass-Antrag des asiatischen Studenten Richard Lee abgelehnt, weil der Computer auf dem Passbild seine Augen nicht erkennen konnte.

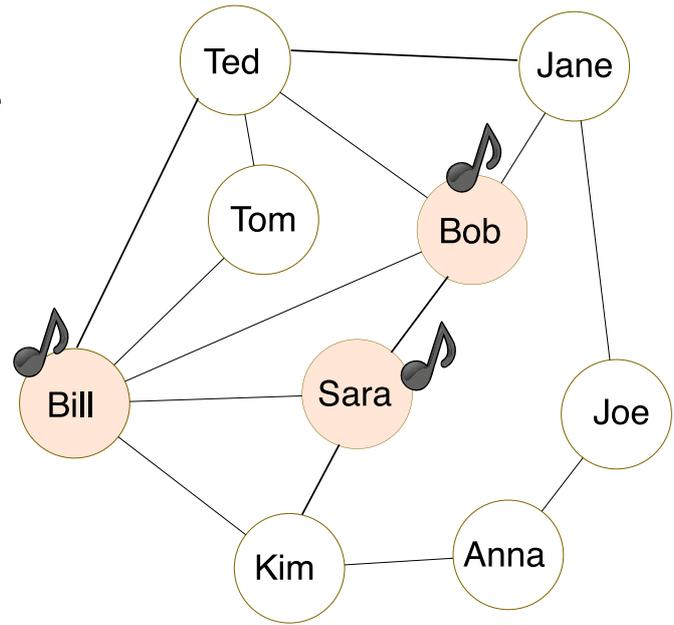


Der neue Song

Das Diagramm zeigt genau, wer in der Klasse mit wem befreundet ist.

Zwei Personen sind Freunde, wenn ihre Namen durch eine Linie verbunden sind; sonst nicht.

Am Montag veröffentlicht ein Megastar seinen neuen Song. Noch am selben Tag kaufen Bill, Bob und Sara den neuen Song. Ihre Namen sind mit einer Note markiert.

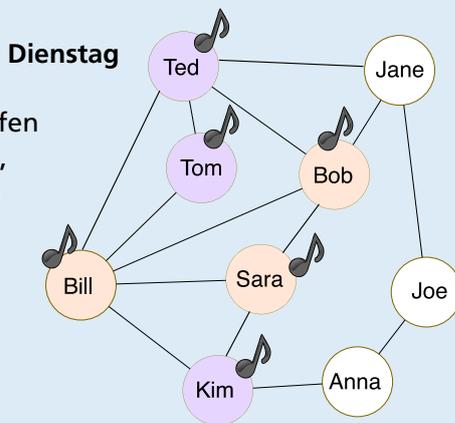


Ab Dienstag passiert jeden Tag Folgendes: Genau die Schüler kaufen den Song, für die gilt, dass mindestens die Hälfte ihrer Freunde den Song schon am Vortag hatte. Am Dienstag kauft also u.a. Tom den Song, Jane aber nicht.

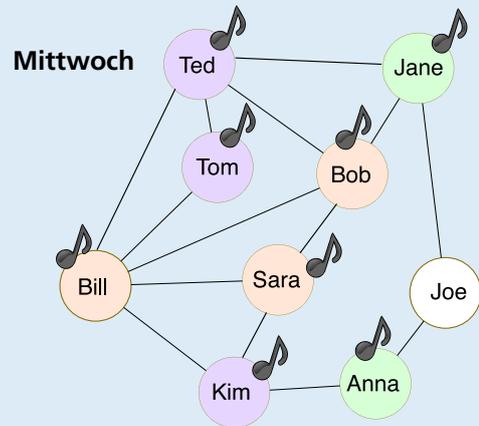
An welchem Tag frühestens haben alle in der Klasse den neuen Song?

- A) Mittwoch
- B) Donnerstag
- C) Freitag
- D) Samstag

Antwort B ist richtig:



Tom, Ted und Kim kaufen den Song am Dienstag, Anna und Jane kaufen ihn am Mittwoch:



Joe kauft daraufhin den Song am Donnerstag. An diesem Tag haben alle in der Klasse den Song; vorher nicht.

Das ist Informatik!

So genannte „Soziale Netzwerke“ haben manchmal Milliarden Mitglieder. Deshalb sind sie nicht nur für ihre BenutzerInnen interessant, sondern vor allem auch für Anbieter von Produkten. In Marketingkampagnen wird der Diffusionsprozess in sozialen Netzwerken ausgenutzt, um neue Produkte bekannt zu machen. Um diesen Prozess geht es in dieser Biberaufgabe.

Hier wird ein Schwellenmodell verwendet, um die Diffusion zu simulieren. Das Diagramm in der Aufgabe ist ein Graph. Er besteht aus Knoten (Namen von Personen) und Kanten (Linien zwischen den Knoten, die Freundschaften – im Sinne des sozialen Netzwerkes – zwischen Personen darstellen). Knoten, die über eine Kante direkt miteinander verbunden sind, nennt man Nachbarn. Die Infektionsschwelle q ist eine Zahl zwischen 0 und 1. In der Aufgabe ist $q = 0,5$. Eine Person zeigt ein neues Verhalten B (z. B. Kauf eines Produktes), wenn mindestens der Anteil q ihrer Nachbarn im Graph auch dieses Verhalten B zeigt.



Ehrenname

Yurine aus Japan gibt ihren Freunden Ehrennamen, nach einer alten Tradition. Ein Ehrenname wird aus dem richtigen Vornamen gemacht, indem jeder Buchstabe des Vornamens durch eine Silbe ersetzt wird, und zwar so:

A → ka	F → lu	K → me	P → mor	U → do	Z → zi
B → pi	G → ji	L → ta	Q → ke	V → ru	
C → mi	H → ri	M → rin	R → shi	W → mei	
D → te	I → ki	N → to	S → ari	X → na	
E → ku	J → zu	O → mo	T → chi	Y → fu	

Einem Freund aus Kroatien gibt Yurine diesen Ehrennamen:
Zukame Moru

Wie lautet der richtige Vorname des Freundes?

- A) Josip
- B) Jani
- C) Jakov
- D) Jurica

Antwort C ist richtig:

Der richtige Vorname des Freundes lautet Jakov.

Man kann den Ehrennamen von vorne durchgehen und zuerst bestimmen, welcher Buchstabe durch die Silbe "Zu" ersetzt wird: "J". Nach der zweiten Silbe "ka" ("A") fallen die Antworten „Josip“ und „Jurica“ bereits weg, und nach der dritten Silbe "me" ("K") kommt nur noch die Antwort „Jakov“ in Frage. Zur Sicherheit könnte man "Moru" noch überprüfen, indem man "O" ("mo") und "V" ("ru") ersetzt. Das geht schneller als nach "mo" und "ru" zu suchen, weil die Tabelle nach den Buchstaben sortiert ist, während die Silben unsortiert sind.

Sehr schnell kann man die richtige Antwort übrigens finden, indem man nach dem zur letzten Silbe "ru" passenden Buchstaben sucht: Das ist das „V“ aus „Jakov“. Ohnehin kommen nur „Josip“ und „Jakov“ in Frage, da der Ehrenname aus fünf Silben besteht.

Das ist Informatik!

Wenn Daten wie etwa die Buchstaben des Alphabets durch andere Daten (wie hier die Silben) ersetzt werden, spricht die Informatik von einer Codierung bzw. einem Code. Es gibt viele Gründe dafür, eine Codierung anzuwenden. Zum Beispiel kann eine passende Codierung die Verarbeitung von Daten erleichtern oder sogar erst ermöglichen: Daten, die mit Hilfe von Computern gespeichert und verarbeitet werden sollen, müssen binär codiert werden, also indem Buchstaben, Ziffern, Bildpunkte, Töne usw. durch Folgen von 0- und 1-Werten ersetzt werden.

Besonders berühmt sind Codes zur Verschlüsselung und damit Geheimhaltung von Information. Da ist es aber keine gute Idee, jedes Zeichen immer wieder auf die gleiche Weise zu ersetzen, so wie in dieser Biberaufgabe bei den Ehrennamen. Bei einer solchen Verschlüsselung handelt es sich nämlich um eine „monoalphabetische Substitution“ (Substitution ist ein Fremdwort für Ersetzung), und die sind besonders leicht zu knacken. Bei den Ehrennamen ist das nicht weiter schlimm, denn es soll ja nichts geheim gehalten werden.



Ersetzungen

Herr Müller und Frau Maier arbeiten im gleichen Projekt. Sie wollen mehr Abwechslung bei der Arbeit und haben vereinbart, ihre Aufgaben miteinander zu tauschen. Ihre Chefin ist einverstanden.

Im Text der Projektbeschreibung sollen nun alle Vorkommen der Zeichenfolge „Müller“ durch die Zeichenfolge „Maier“ ersetzt werden – und alle „Maier“ durch „Müller“.

In der Textverarbeitung können beliebige Zeichenfolgen durch andere ersetzt werden. Im Text kommt das Zeichen „#“ nicht vor.

Wie gelingt der Austausch von „Müller“ und „Maier“?

- A) Ersetze zuerst alle „Müller“ durch „Maier“ und dann alle „Maier“ durch „Müller“.
- B) Ersetze zuerst alle „Maier“ durch „Müller“ und dann alle „Müller“ durch „Maier“.
- C) Ersetze zuerst alle „Müller“ durch „#“, dann alle „#“ durch „Maier“ und dann alle „Maier“ durch „Müller“.
- D) Ersetze zuerst alle „Müller“ durch „#“, dann alle „Maier“ durch „Müller“ und dann alle „#“ durch „Maier“.

Antwort D ist richtig:

A: Hier würden nur „Müller“ übrig bleiben und alle „Maier“ verloren gehen: Nach der ersten Ersetzung kommen nur noch „Maier“ vor, die alle durch „Müller“ ersetzt werden.

B: Hier würden nur „Maier“ übrig bleiben und alle „Müller“ verloren gehen: Nach der ersten Ersetzung kommen nur noch „Müller“ vor, die alle durch „Maier“ ersetzt werden.

C: Hier würden nur „Müller“ übrig bleiben und alle „Maier“ verloren gehen: Nachdem alle „Müller“ durch „#“ ersetzt wurden, werden diese sofort weiter durch „Maier“ ersetzt, die wiederum alle durch „Müller“ ersetzt werden.

D: Indem zunächst alle „Müller“ durch „#“ ersetzt werden, bleibt erhalten, wo „Müller“ ursprünglich vorkam – im Gegensatz zu den „Müller“, die durch die darauf folgende Ersetzung von „Maier“ durch „Müller“ entstehen. Durch die abschließende Ersetzung von „#“ durch „Maier“ werden nur die ursprünglichen „Müller“ durch „Maier“ ersetzt.

Das ist Informatik!

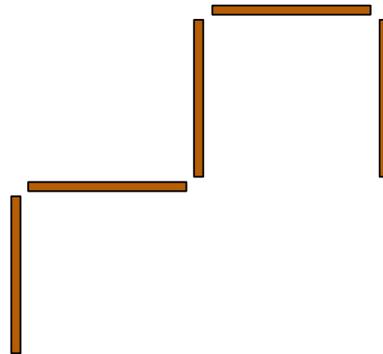
Obwohl eine einzelne Ersetzung ein ganz einfacher Vorgang ist, spielen Ersetzungen eine wichtige Rolle in der Informatik. Mittels Serien von Ersetzungen können komplexe Aufgaben durchgeführt werden. Auch in der Definition der Syntax von Programmiersprachen mit Hilfe formaler Grammatiken kommen sie vor. Eine formale Grammatik wird im Wesentlichen als eine Liste von Ersetzungsregeln formuliert. In dieser Aufgabe besteht die Schwierigkeit darin, dass zwei Zeichenfolgen miteinander vertauscht werden sollen. Dies funktioniert nur, wenn eine dritte, sonst nicht verwendete Zeichenfolge temporär verwendet wird.

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

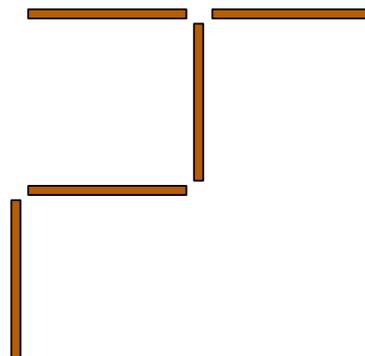


Fünf Hölzchen

Fünf Hölzchen liegen so auf dem Tisch:



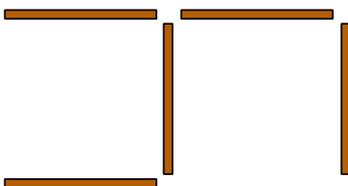
Nola nimmt ein Hölzchen und legt es woanders hin. Jetzt liegen die Hölzchen so:



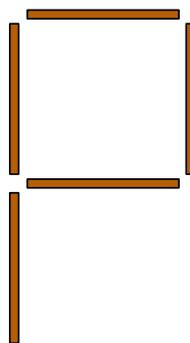
Danach nimmt Bert ein Hölzchen und legt es woanders hin.

Wie können die Hölzchen jetzt **NICHT** liegen?

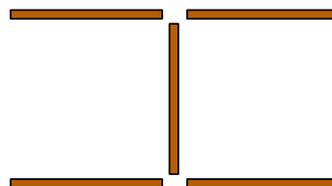
A)



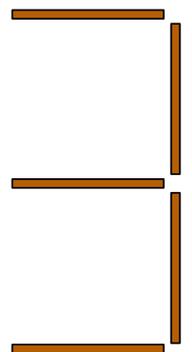
B)



C)

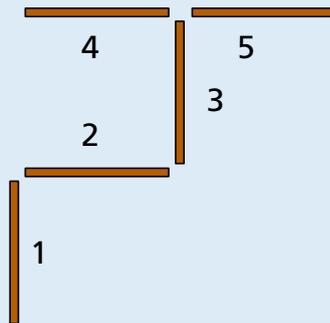


D)



**Antwort D ist richtig:**

Wir geben den Hölzchen Nummern. Nach Nola liegen die Hölzchen so:



Damit die Hölzchen so wie in **Bild A** liegen, muss Bert das Hölzchen 1 woanders hin legen.

Damit die Hölzchen so wie in **Bild B** liegen, muss Bert das Hölzchen 5 woanders hin legen.

Damit die Hölzchen so wie in **Bild C** liegen, muss Bert das Hölzchen 1 woanders hin legen.

Damit die Hölzchen so wie in **Bild D** liegen, müsste Bert zwei Hölzchen, nämlich die Hölzchen 1 und 5 woanders hin legen. Er legt aber nur ein Hölzchen woanders hin.

Das ist Informatik!

Um die Lage der Hölzchen zu verändern, haben Nola und Bert die einfache Möglichkeit, genau ein Hölzchen woanders hinzulegen. Damit verändert sich die Lage der Hölzchen so wenig wie möglich. Wenn ein Freund ihnen sagen möchte, wie sie die Lage der Hölzchen verändern sollen, kann er die folgende Anweisung ein- oder mehrmals benutzen: „Lege ein Hölzchen woanders hin!“ – wobei er auch noch sagen muss, welches Hölzchen und wohin. Weitere Anweisungen, z.B. „Lege zwei Hölzchen woanders hin!“, „Lege drei Hölzchen woanders hin!“ usw. werden nicht benötigt.

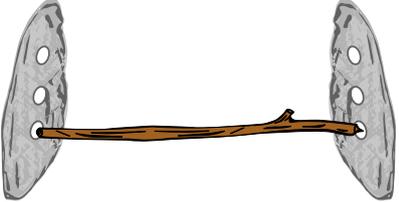
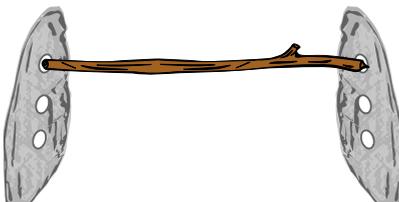
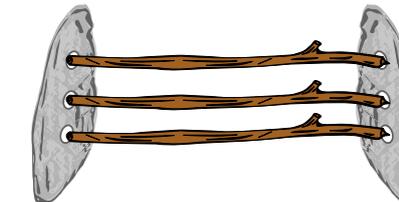
Wer Computer programmiert, schreibt sogenannte Anweisungen auf, die der Computer schon kennt und daher ausführen kann. Kompliziertere Anweisungen werden aus den einfachen Anweisungen zusammengesetzt: indem man entweder mehrere Anweisungen hintereinander aufschreibt, oder indem man aufschreibt, dass eine Anweisung mehrfach hintereinander ausgeführt werden soll, oder indem man aufschreibt, dass eine Anweisung nur unter einer bestimmten Bedingung ausgeführt werden soll. Das sind die drei wichtigsten Möglichkeiten, in Computerprogrammen aus (anfänglich einfachen) Anweisungen immer kompliziertere Anweisungen zu machen. Das hört sich doch gar nicht so schwer an!



Gartentor

Die Biber besuchen einander oft. Aber ... manchmal ist man nicht zu Hause. Dann hinterlässt man an seinem Steinplatten-Gartentor eine Nachricht, wann man ungefähr wieder da ist. Dazu steckt man bis zu drei Holzstäbe gerade zwischen gegenüberliegende Bohrlöcher der Steinplatten.



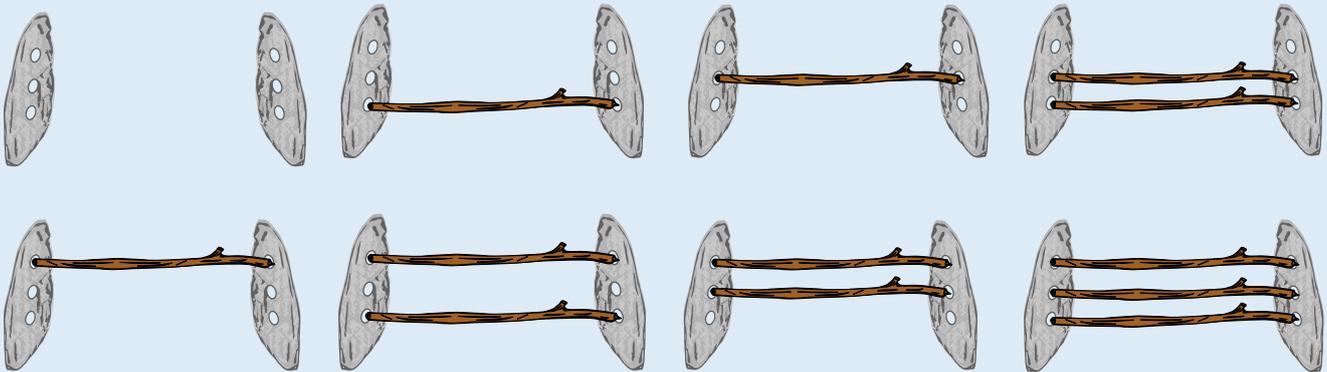
 <p>Wir sind da, kommt doch rein.</p>	 <p>Wir werden schon mittags zurück sein.</p>
 <p>Wir sind leider erst am Abend wieder da.</p>	 <p>Wir besuchen jemand und kommen um Mitternacht heim.</p>

Die Biber haben diese vier Gartentor-Nachrichten miteinander verabredet: Die Biber könnten noch weitere Nachrichten verabreden, ohne zusätzliche Holzstäbe oder Bohrlöcher zu verwenden.

Wie viele?



4 ist die richtige Antwort:



Dies sind alle acht Nachrichten, die man mit bis zu drei Holzstäben in die Bohrlöcher stecken kann. Bei vier Nachrichten ist deren Bedeutung schon verabredet, bei den anderen vier Nachrichten noch nicht.

Das ist Informatik!

In dieser Biberaufgabe benutzen die Biber ein dreistelliges binäres Zeichensystem. Die Träger der Information sind die gegenüberliegenden Bohrloch-Paare. Ein solches Bohrloch-Paar hat zwei Zustände, und zwar „Holzstab gesteckt“ oder „Holzstab nicht gesteckt“. Die Anzahl der Bohrloch-Paare gibt die Stelligkeit wieder. Mathematisch gilt: Die Anzahl unterschiedlicher Nachrichten ist gleich der Anzahl der Bohrloch-Paar-Zustände (2) hoch der Anzahl der Bohrloch-Paare (3). Also: $2 \text{ hoch } 3 = 2 \cdot 2 \cdot 2 = 8$.

Die Biber wissen aus Tradition, was ihre freundlichen Nachrichten bedeuten, und wenn sie sich mal irren, kann kein großer Schaden entstehen. In der Informatik mit ihren global vernetzten Systemen sind dagegen alle mit allen direkte Kommunikationsnachbarn und müssen sich fehlerfrei verstehen.

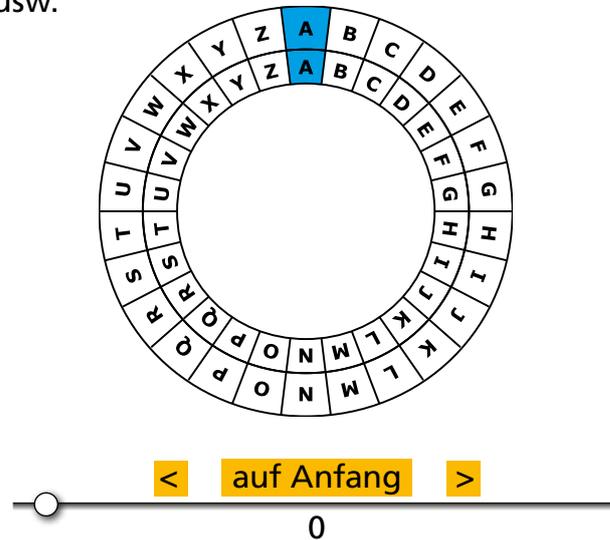
Große Organisationen kümmern sich um die Standardisierung von Zeichensystemen und die Zertifizierung von Produkten. In multinational besetzten Fachausschüssen wird über das Aussehen von Zeichen und deren Bedeutung verhandelt und beschlossen. Die wichtigsten Zeichensysteme werden von Parlamenten vieler Staaten zu Gesetzen erklärt. So sorgen sich viele Menschen in aller Welt darum, dass sich alle Computer der Welt gegenseitig gut verstehen.

https://de.wikipedia.org/wiki/Normung#Internationale_Normung



Geheime Bestellung

Anna bestellt im Restaurant mit geheimen Botschaften; nur Cäsar der Koch soll sie verstehen. Sie benutzt dazu ihre Geheimscheibe. Diese hat einen äußeren und einen inneren Ring mit den Buchstaben des Alphabets. Zu Anfang stehen die Ringe gleich: A (innen) passt zu A (außen), B passt zu B usw.



So erstellt Anna eine geheime Botschaft: Zuerst schreibt sie ihre Bestellung auf, z.B. PIZZA. Dann macht sie für jeden Buchstaben:

1. Unter den Buchstaben schreibt sie eine "Dreh-Zahl".
2. Sie stellt den inneren Ring auf Anfang und dreht ihn dann um so viele Buchstaben nach links, wie die Dreh-Zahl angibt.
3. In die Botschaft schreibt sie den Buchstaben, der nun zum Buchstaben aus der Bestellung passt.

Ein Beispiel: Wenn Anna PIZZA haben will und die Dreh-Zahlen 3, 1, 4, 3 und 1 benutzt, erstellt sie die geheime Botschaft SJDCB. Im Bild siehst du, wie die Ringe bei Dreh-Zahl 3 stehen; S passt nun zu P.

Bestellung	P	I	Z	Z	A
Dreh-Zahlen	3	1	4	3	1
Botschaft	S	J	D	C	B

Für eine andere Bestellung hat Anna mit den Dreh-Zahlen 3, 5, 1, 7, 2, 4 und 8 die Botschaft OFTHIRI erstellt.

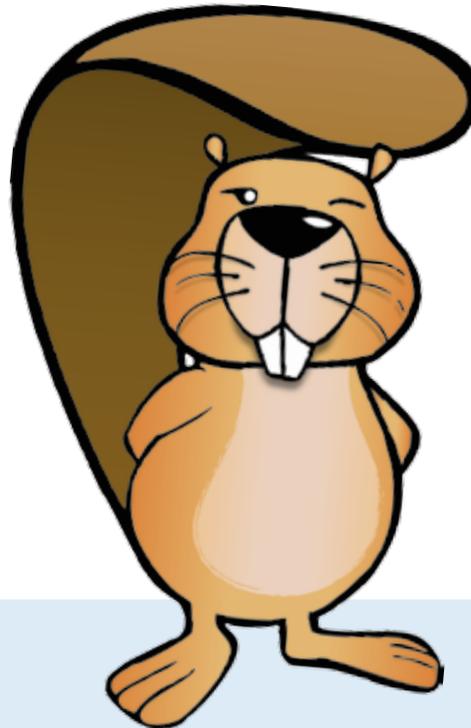
Wie lautet die Bestellung?

Schreibe die Buchstaben in die freien Felder. Zur Hilfe kannst du den inneren Ring drehen.

Bestellung	?	?	?	?	?
Dreh-Zahlen	3	2	1	3	2
Botschaft	S	C	T	W	C



Du liest jetzt schon die zwölfte Biberaufgabe
 in diesem Heft. Zielgruppe: Informatik-interessiert.
 Du solltest Programmieren lernen:
wettbewerb.jwinf.de
cscircles.cemc.uwaterloo.ca/de



So ist es richtig:

Bestellung	P	A	S	T	A
Dreh-Zahlen	3	2	1	3	2
Botschaft	S	C	T	W	C

Die Bestellung bekommt man mit Hilfe der Geheimscheibe heraus, indem man für jeden Buchstaben der Botschaft den inneren Ring gemäß der Dreh-Zahl nach links dreht. Nun sucht man den Buchstaben der Botschaft auf dem inneren Ring. Der Buchstabe der Bestellung ist dann der passende Buchstabe auf dem äußeren Ring.

Das ist Informatik!

Anna verschlüsselt ihre Bestellungen, damit nur ihr Lieblingskoch sie versteht. Verschlüsselung ist eines der ältesten Anliegen der Menschheit. Schon immer hat es Gründe dafür gegeben, Botschaften so zu übermitteln, dass nur die gewünschten Empfänger sie verstehen können. Es gibt viele verschiedene Verschlüsselungsverfahren. Aber immer gehören zwei Algorithmen dazu, nämlich einer zum Verschlüsseln und einer zum Entschlüsseln, und beide benötigen für ihre Arbeit den zur Botschaft gehörigen Schlüssel.

Eines der einfachsten Verschlüsselungsverfahren geht auf Julius Cäsar zurück: Hier ist der Schlüssel eine Zahl, die eine Verschiebung im Alphabet angibt. Der Schlüssel 3 bedeutet z.B., dass der Buchstabe A einer Nachricht mit D verschlüsselt wird, B mit E usw. – und dass der Buchstabe D als A zu entschlüsseln ist, E als B usw. Beim Verschlüsseln und Entschlüsseln nach dieser Methode hilft die „Cäsar-Scheibe“, die in dieser Biberaufgabe beschrieben ist.

Verschlüsselungsverfahren, die für eine Botschaft nur einen Schlüssel verwenden, sind unsicher. Das weiß Anna anscheinend, denn sie benutzt für jeden Buchstaben einen anderen Schlüssel – ganz ähnlich wie beim Verfahren von Vigenère. In diesem Verfahren wiederholen sich die Dreh-Zahlen bei längeren Botschaften; so wird der Schlüssel nicht zu lang.



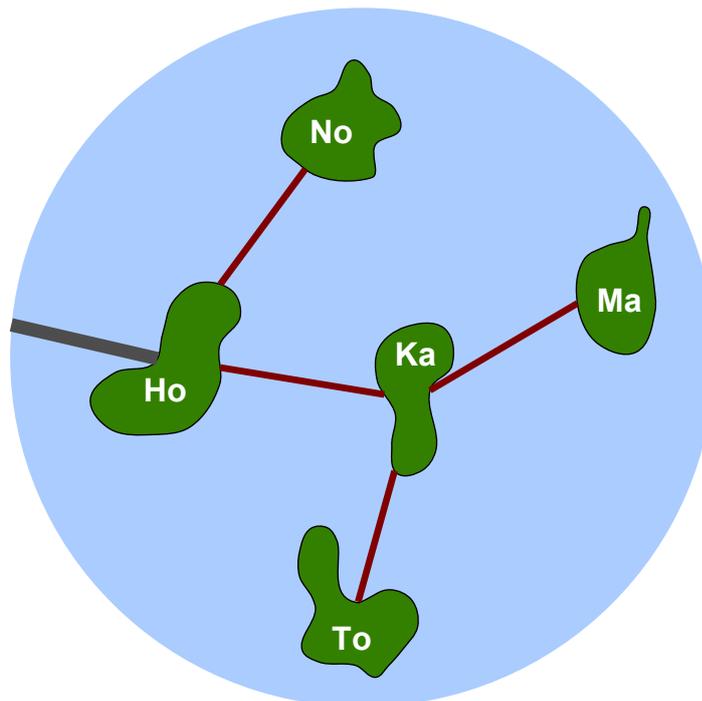
Honomakato

Die Inselgruppe Honomakato besteht aus den fünf Inseln Ho, No, Ma, Ka und To.

Die Hauptinsel Ho ist über ein Glasfaserkabel mit dem Internet verbunden.

Außerdem sind einige Kupferkabel verlegt: zwischen Ho und No, Ho und Ka, Ka und Ma sowie Ka und To.

So sind alle Inseln mit Ho verbunden und dadurch auch mit dem Internet.



verbinden

trennen

Die Bewohner von Honomakato möchten eine robuste Verbindung aller Inseln mit dem Internet: Auch wenn irgendeines der Kupferkabel beschädigt würde, sollte jede Insel noch mit dem Internet verbunden sein.

**Sorge dafür, dass Honomakato eine robuste Verbindung mit dem Internet bekommt.
Verlege dazu zwei weitere Kupferkabel zwischen den Inseln.**

Um zwischen zwei Inseln ein Kabel zu verlegen, klicke auf die beiden Inseln und dann auf „verbinden“. Klicke auf „trennen“, um ein Kabel wieder zu entfernen.
Es gibt mehrere richtige Antworten.

**So ist es richtig:**

Durch das Verlegen zweier weiterer Kabel kann dafür gesorgt werden, dass Honomakato eine robuste Verbindung mit dem Internet bekommt. Dabei gibt es sechs verschiedene Möglichkeiten. Die einzelnen Kabel schützen jeweils bestimmte Inseln davor, durch die Beschädigung eines Kabels nicht mehr mit dem Internet verbunden zu sein.

Ho-Ma und No-To: Ho-Ma schützt Ma and Ka, No-To schützt No and To.

Ho-To und No-Ma: Ho-To schützt To and Ka, No-Ma schützt Ma, No, and Ka.

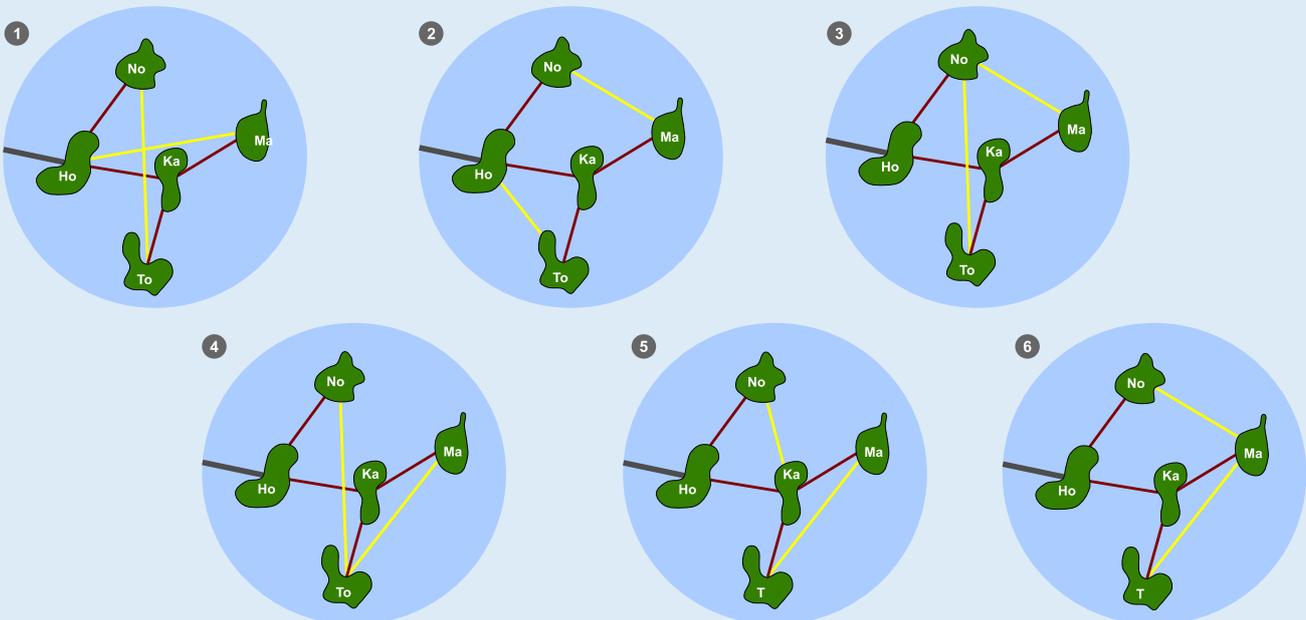
No-To und No-Ma: No-To schützt No, To and Ka, No-Ma schützt Ma, No, and Ka.

No-To und Ma-To: No-To schützt No, To and Ka, Ma-To schützt Ma and To.

No-Ka und Ma-To: No-Ka schützt No and Ka, Ma-To schützt Ma and To.

No-Ma und Ma-To: No-Ma schützt Ma, No, and Ka, Ma-To schützt Ma and To.

Für die Lösungen gilt jeweils: Jede Insel ist mit mindestens zwei Verbindungen ausgestattet, und Honomakato lässt sich nicht in zwei Teile gruppieren, zwischen denen nur eine Verbindung existiert.

**Das ist Informatik!**

Das Kabelnetzwerk, mit dem die Inseln von Honomakato mit dem Internet verbunden wurden, ist einerseits ein kleiner Teil des Internet. Andererseits ist sein Aufbau aber auch ein Beispiel für den Aufbau des gesamten Internet. Die Router, Server und anderen Geräte mit fester Internetadresse sind Knoten im großen Netzwerk „Internet“, genauso wie hier die Inseln im Netzwerk von Honomakato.

Das Internet wurde in den 1960er Jahren als robustes Netzwerk erfunden. Der Ausfall von Verbindungen zwischen einzelnen Netzwerkknoten sollte nicht zum Ausfall des gesamten Netzwerks führen. Daher verbindet man die Knoten mehrfach miteinander und konfiguriert sie so, dass beim Ausfall oder der Überlastung einer Leitung einfach eine andere Leitung verwendet wird. Auch für andere Netzwerkarten wie Verkehrsnetze oder Versorgungsnetzwerke ist es wichtig, dass es keine einzelne Verbindung oder auch einzelnen Knoten gibt, deren Ausfall ein komplettes Netzwerk außer Gefecht setzt.

Die Informatik setzt Graphentheorie ein, um Berechnungen über Netzwerke durchzuführen. Ein Graph ist (theoretisch) ein Netzwerk aus Knoten und Verbindungen (Kanten genannt) zwischen Knoten. Ein Graph heißt „zusammenhängend“, wenn für jedes Paar von Knoten A und B gilt, dass B mit A durch einen Weg über eine oder mehrere Kanten verbunden ist. Eine Kante in einem Graphen, die vorhanden sein muss, damit der Graph zusammenhängend ist, wird als Brücke bezeichnet. Die Informatik kennt Verfahren, um Brücken in Graphen zu finden. Von Robert Tarjan stammt ein solcher (effizienter) Algorithmus – sowie viele andere Algorithmen auf Graphen.

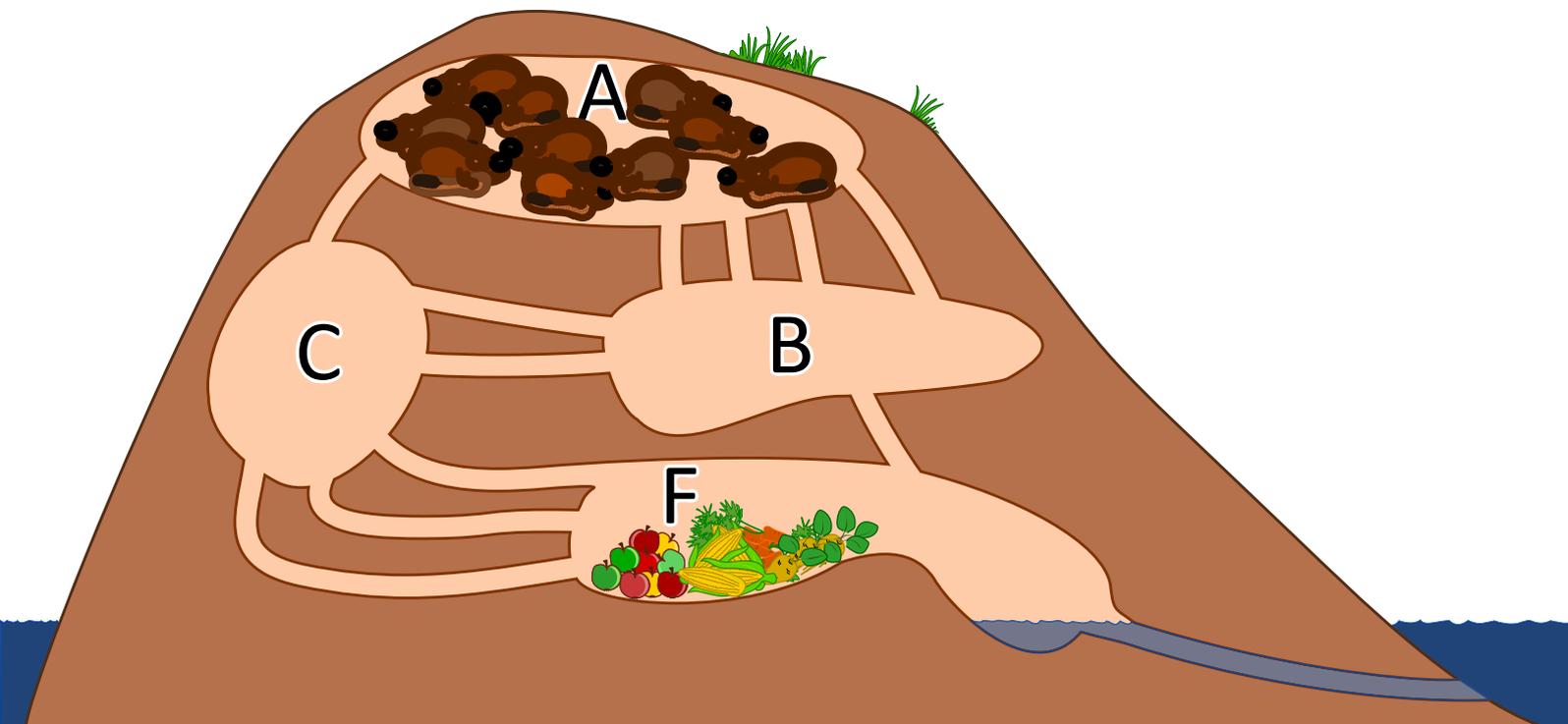


Hunger!

10 Biber befinden sich in Raum A ihrer Burg. Sie 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. Es können also nicht mehrere Biber direkt hintereinander durch einen Gang laufen. In den Räumen A, B, C und F ist genug Platz für alle Biber, und das Durchqueren eines Raums kostet keine Zeit.

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



4 ist die richtige Antwort:

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 dann insgesamt erst 6 Biber in Raum F. Wir brauchen also noch eine 4. Minute. Wie z.B. alle 10 Biber in 4 Minuten ans Ziel kommen können, zeigt folgende Tabelle:



Aktion / Situation	Anzahl Biber in den Räumen (nach der Aktion)			
	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 Möglichkeiten, wie die Biber innerhalb von 4 Minuten in Raum F kommen können. Bei der hier gezeigten Möglichkeit 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 wieviele Biber innerhalb einer Minute von einem zum anderen Raum gehen können. Es ist die sogenannte Kapazität der Verbindung zwischen zwei Räumen, die 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.



Klammerschmuck

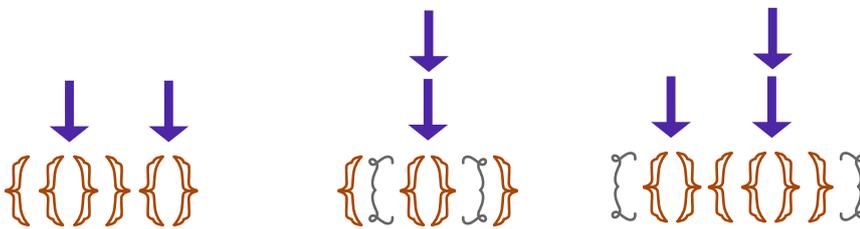
Die Firma „Klammerschmuck“ stellt Armbänder aus klammerförmigen Spangen her. Die Firma verwendet die Spangen immer paarweise, und immer in der gleichen Anordnung.

Die Herstellung eines Armbands beginnt mit einem dieser Paare:

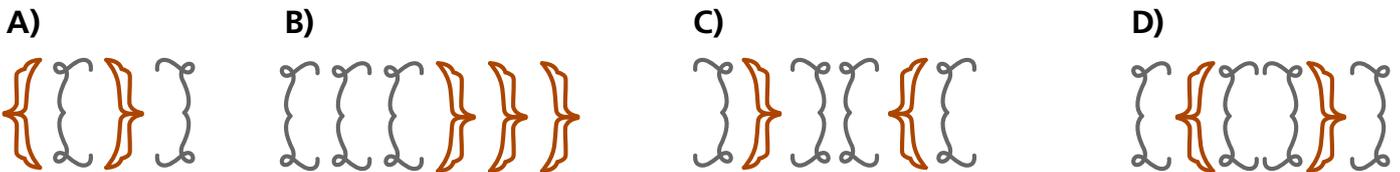


Danach werden wiederholt weitere Paare an beliebigen Stellen eingefügt.

Hier sind Beispiele für drei verschiedene Armbänder:



Welches der folgenden Armbänder wurde von der Firma „Klammerschmuck“ hergestellt?



Antwort D ist richtig:

Die Herstellung dieses Armbands hat mit dem Paar $\{\}$ begonnen. In dieses Paar wurde ein Paar $\{\}$ eingefügt und in dieses wiederum ein weiteres Paar $\{\}$.

Alle anderen Armbänder wurden nicht von der Firma „Klammerschmuck“ hergestellt:

- Das Armband A besteht aus zwei Paaren. Egal mit welchem begonnen wurde, das zweite Paar wurde nicht als Paar in das erste eingefügt, sondern aufgeteilt.
- Bei Armband B öffnen sich die Spangen nach außen, nicht nach innen wie bei der Firma „Klammerschmuck“.
- Das Armband C besteht überhaupt nicht aus Paaren gleicher Spangen.

Da ist Informatik!

Die Armbänder der Firma „Klammerschmuck“ bestehen aus Klammerspaaren, bei denen sich die linke Klammer nach rechts öffnet und die rechte Klammer nach rechts schließt. Solche öffnenden und schließenden Klammern sind aus der Mathematik bekannt: die runden Klammern (und), die geschweiften Klammern { und } sowie die eckigen Klammern [und]. Auch in der Informatik gibt es Klammern. In vielen Programmiersprachen wie etwa Java und C++ werden zusammenhängende Programmteile mit geschweiften Klammern kenntlich gemacht. Mit eckigen Klammern kann u.a. in der Programmiersprache Python eine Reihe von Werten zu einer Liste zusammengefügt werden. In der Webseiten-Sprache HTML wiederum sind die spitzen Klammern < und > besonders wichtig.

Mehr zur Bedeutung von Klammern in der Informatik ist bei der Aufgabe „BikeFun“ zu finden.

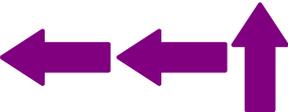


Kreisch!

Elena und Dustin nehmen an einer Tanzshow teil.

Je nach Reaktion des Publikums führen sie gleichzeitig bestimmte Tanzschritte aus.

Die Tabelle zeigt, wie sie sich über die Felder der Tanzfläche bewegen:

	Wow!	Kreisch!	Klatsch!	Buh!
Elena 				
Dustin 				

Wenn zum Beispiel das Publikum kreischt, bewegt Elena sich um ein Feld nach oben und danach um ein Feld nach links; gleichzeitig bewegt sich Dustin um ein Feld nach rechts und danach um ein Feld nach unten.

Elena und Dustin starten auf der Tanzfläche auf diesen Feldern:

Das Publikum möchte, dass die beiden am Ende auf dem gleichen Feld stehen.

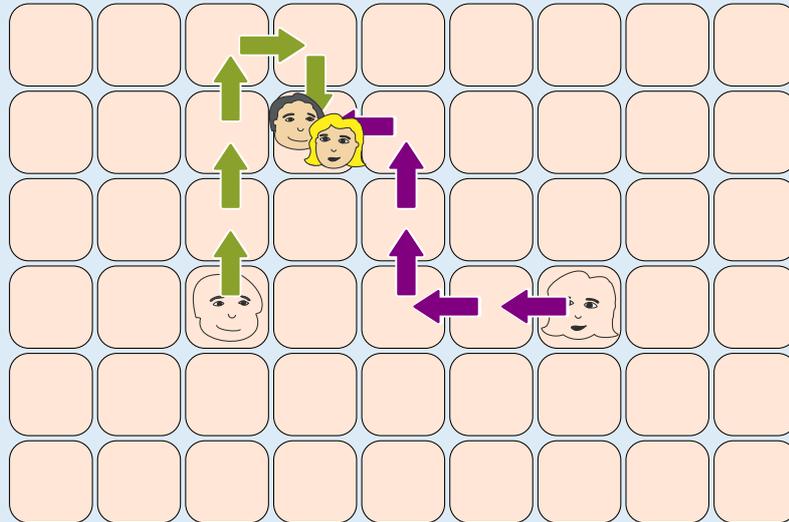
Welche Reaktionen muss das Publikum dazu zeigen?

- A) Buh! Kreisch!
- B) Wow! Kreisch!
- C) Kreisch! Kreisch!
- D) Klatsch! Kreisch!

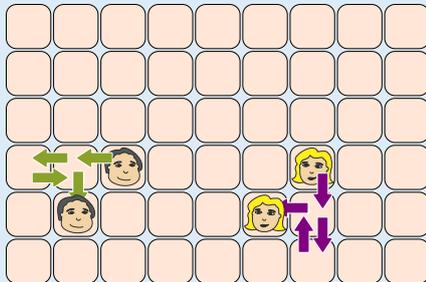
**Antwort D ist richtig:**

Bei „Klatsch! Kreisch!“ bewegt Elena sich um insgesamt zwei Felder nach oben und drei Felder nach links. Dustin bewegt sich insgesamt um ebenfalls zwei Felder nach oben und um ein Feld nach rechts. Beim Start waren beide auf gleicher Höhe und vier Felder auseinander. Am Ende sind sie wieder auf gleicher Höhe und haben sich um insgesamt vier Felder aufeinander zu bewegt. Sie stehen dann auf dem gleichen Feld. Das Bild zeigt die Bewegungen:

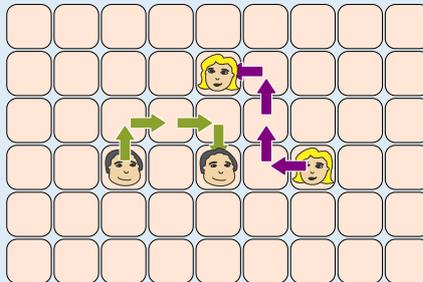


In allen anderen Fällen stehen Dustin und Elena am Ende nicht auf dem gleichen Feld:

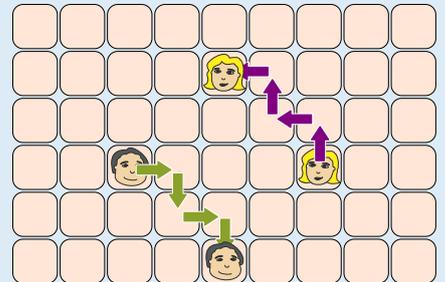
Antwort A: Buh! Kreisch!



Antwort B: Wow! Kreisch!



Antwort C: Kreisch! Kreisch!

**Das ist Informatik!**

In dieser Biberaufgabe handeln zwei Akteure, nämlich Dustin und Elena (oder waren es Justin und Selena?) unabhängig voneinander. Dabei sollen die Akteure eines ihrer Arbeitsmittel (ein Feld auf der Tanzfläche) gemeinsam benutzen. Das wird wohl gut gehen, denn die beiden haben sicher nicht zum ersten Mal an einer Tanzshow teilgenommen.

Die handelnden Akteure beim Betrieb eines Computers nennt die Informatik Prozesse. Auch sie sind grundsätzlich voneinander unabhängig. Die Arbeitsmittel (auch Ressourcen genannt) der Prozesse sind z.B. der Arbeitsspeicher des Computers oder an den Computer angeschlossene Geräte. Wenn zwei Prozesse zur gleichen Zeit dasselbe Arbeitsmittel benutzen wollen, geht das ohne Weiteres nicht gut. Wenn etwa zwei Prozesse die gleichen Bereiche im Arbeitsspeicher beschreiben dürften gerieten alle ihre Berechnungen durcheinander.

Die Informatik hat deshalb viele Mechanismen entwickelt, um die Vergabe von Arbeitsmitteln an die Prozesse im Computer zu regeln. Umgesetzt sind sie im zentralen Steuerungssystem des Computers, dem Betriebssystem.



Kreiselstadt

In Kreiselstadt gibt es nur Kreisverkehre und sonst keine anderen Kreuzungen.
An vier Stellen A, B, C und D kann man in die Stadt hineinfahren.
Unten siehst du einen Plan.

Wenn die Kreiselstädter einen Weg beschreiben, sagen sie zum Beispiel:

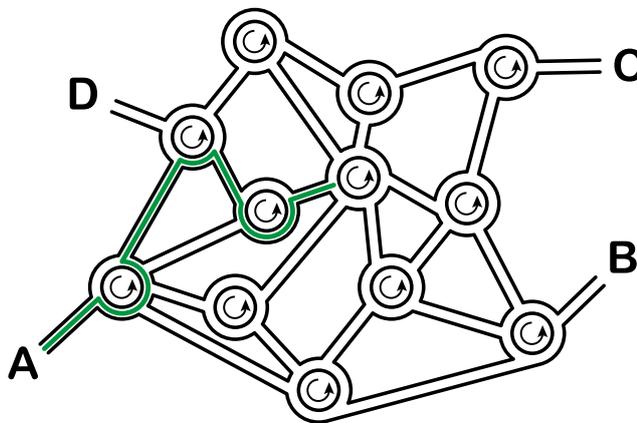
Beim nächsten Kreisverkehr nimm die 4. Ausfahrt.

Beim dann folgenden Kreisverkehr nimm die 1. Ausfahrt.

Beim dann folgenden Kreisverkehr nimm die 2. Ausfahrt.

Einem einheimischen Fahrer sagen sie stattdessen nur kurz die Wegbeschreibung „4 1 2“.

Von A aus führt diese Wegbeschreibung zu einem Kreisverkehr in der Stadtmitte:

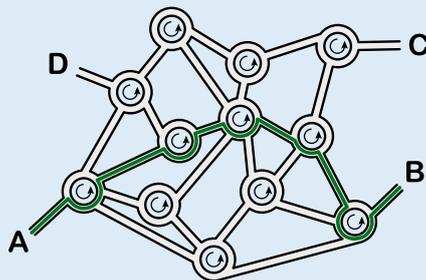


Wohin führt die Wegbeschreibung „3 1 3 2 3“ von A aus?

- A) Sie führt zu A. B) Sie führt zu B.
C) Sie führt zu C. D) Sie führt zu D.

Antwort B ist richtig:

Wenn man von A aus der Wegbeschreibung folgt, erreicht man schließlich B.



Das ist Informatik!

Diese Biberaufgabe liefert ein gutes Beispiel für formalisierte Information. Ein Steuerungscomputer kann mit einer ausformulierten Wegbeschreibung nicht viel anfangen. Wenn man aber eine Wegbeschreibung wie im Beispiel als Folge von Zahlen angeben kann, dann ist die Information in einer Form, die ein Computer als Folge von Befehlen interpretieren kann. So eine Befehlsfolge nennt man auch Sequenz. Die Sequenz ist ein wichtiger Baustein in fast allen Programmiersprachen.

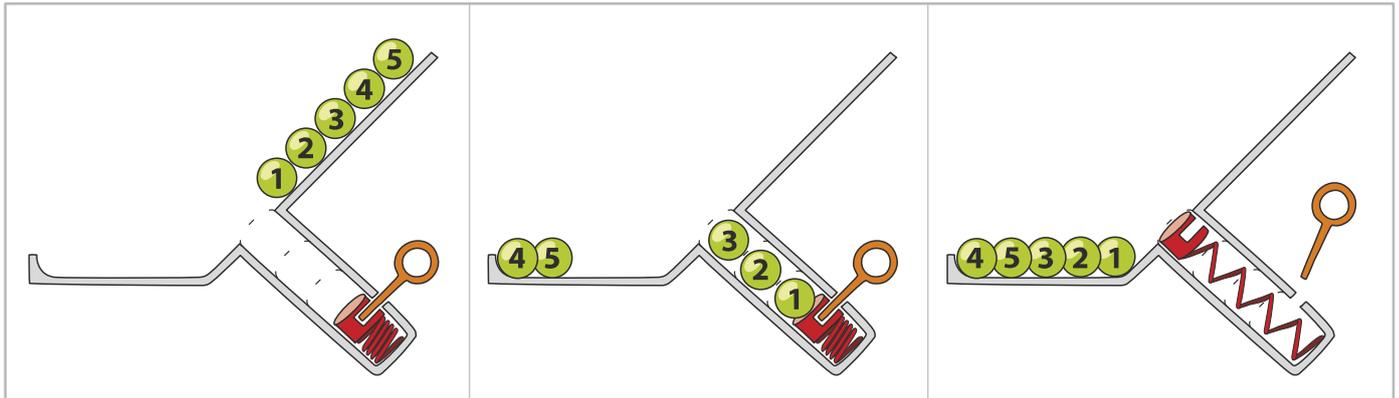
In diesem Beispiel ist es hilfreich, dass das Straßennetz gleichförmig aufgebaut ist: Alle Kreuzungen sind Kreisverkehre. Solche gleichförmigen Strukturen nennt man homogen, im Gegensatz zu gemischten Strukturen, die als heterogen bezeichnet werden. Homogene Strukturen sind für Programme leichter abzuarbeiten als heterogene, weshalb man sie in der Informatik bevorzugt, wenn es keinen zwingenden Grund dagegen gibt.



Kugelrampe

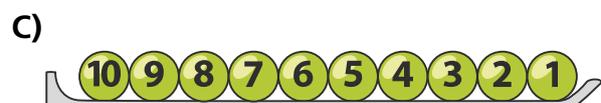
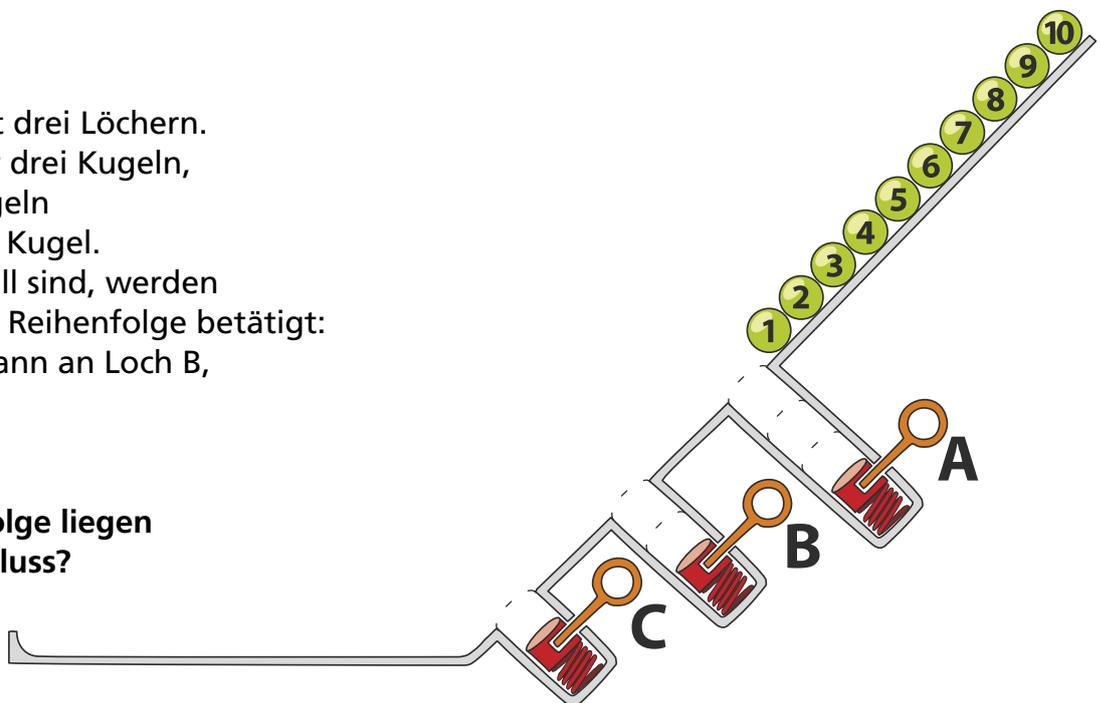
Eine Spielzeugfirma baut Kugelrampen: Nummerierte Kugeln rollen einen Abhang mit Löchern hinunter. Wenn eine Kugel zu einem Loch kommt, fällt sie hinein, wenn noch Platz ist; sonst rollt sie über das Loch. Die Kugeln in einem Loch werden mit einer Feder wieder hinausbefördert – aber nur, wenn gerade keine Kugel rollt. Zum Schluss liegen alle Kugeln am Ende der Rampe.

Hier ist eine Rampe für fünf Kugeln mit einem Loch. Das Loch hat Platz für drei Kugeln. Hier liegen die Kugeln zum Schluss in der Reihenfolge 4, 5, 3, 2, 1 am Ende der Rampe.



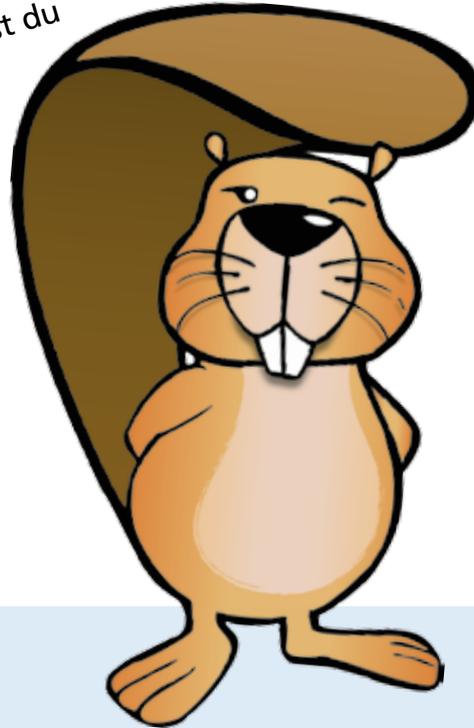
Hier ist eine Rampe für zehn Kugeln mit drei Löchern. Loch A hat Platz für drei Kugeln, Loch B für zwei Kugeln und Loch C für eine Kugel. Wenn die Löcher voll sind, werden die Federn in dieser Reihenfolge betätigt: zuerst an Loch A, dann an Loch B, dann an Loch C.

In welcher Reihenfolge liegen die Kugeln zum Schluss?





Auch Informatik ist ein Teamsport.
 Beim Informatik-Biber dürfen Zweiertteams
 mitmachen. Solche Biber-Teams kannst du
 im Biber-Video sehen:
bwinf.de/biber



Antwort D ist richtig:



Die Kugeln 1, 2 und 3 fallen in Loch A, die Kugeln 4 bis 10 rollen über sie hinweg. Kugeln 4 und 5 fallen in Loch B und Kugeln 6 bis 10 rollen über sie hinweg. Schließlich fällt Kugel 6 in Loch C und die Kugeln 7 bis 10 rollen in dieser Reihenfolge bis an das Ende der Rampe. Nach Lösen der Feder in Loch A werden die Kugeln in der Reihenfolge 3, 2, 1 hinausgeschoben und rollen weiter bis an das Ende der Rampe. Nun sind die Kugeln in der Reihenfolge 7, 8, 9, 10, 3, 2, 1 am Ende der Rampe. Dann wird die Feder in Loch B gelöst, sodass 5 und 4 an das Ende rollen. Zuletzt wird Kugel 6 aus Loch C hinausbefördert. Dadurch sind die Kugeln zum Schluss in der Reihenfolge 7, 8, 9, 10, 3, 2, 1, 5, 4, 6 am Ende der Rampe angekommen.

Das ist Informatik!

Die Löcher in dieser Biberaufgabe funktionieren nach dem Prinzip Last-In First-Out (LIFO): Die letzte Kugel, die hineinfällt, wird zuerst wieder ausgeworfen. Nach dem gleichen Prinzip funktioniert die Datenstruktur Stapel (engl. stack): das zuletzt gespeicherte Daten-Element wird zuerst wieder entfernt.

Auch wenn dieses Prinzip sehr einfach erscheint, ist es doch in vielen Situationen sehr nützlich. Will man beispielsweise automatisch feststellen, ob die Klammern in einem arithmetischen Ausdruck richtig gesetzt sind (in „ $((1 + 2) * 3)$ “ sind sie richtig, in „ $((4 + 5) * (6 - 7))$ “ sind sie falsch), geht man von links nach rechts so vor: Eine öffnende Klammer wird im Stack gespeichert (mit der Operation „push“); wenn eine passende schließende Klammer gefunden wird, wird die öffnende Klammer wieder aus dem Stack entfernt (mit der Operation „pop“). Wird eine schließende Klammer gefunden, ohne dass eine öffnende Klammer im Stack ist, ist ein Fehler im Ausdruck. Ist der Stapel am Ende leer, gab es zu jeder öffnenden Klammer auch eine passende schließende Klammer.

Mehr zum Zusammenhang zwischen der Datenstruktur Stapel und Klammern in der Informatik ist bei der Aufgabe „BikeFun“ zu finden.



Kurzes Programm

Der dreieckige Roboter soll vom roten Startfeld zum grünen Zielfeld gehen.

Der Roboter kann nur sehr kurze Programme verarbeiten.

Programmiere den Roboter!

Ziehe dazu Programmbausteine nach rechts und füge sie zusammen.

Du darfst höchstens vier Bausteine für dein Programm benutzen.

Das bedeuten die Knöpfe:



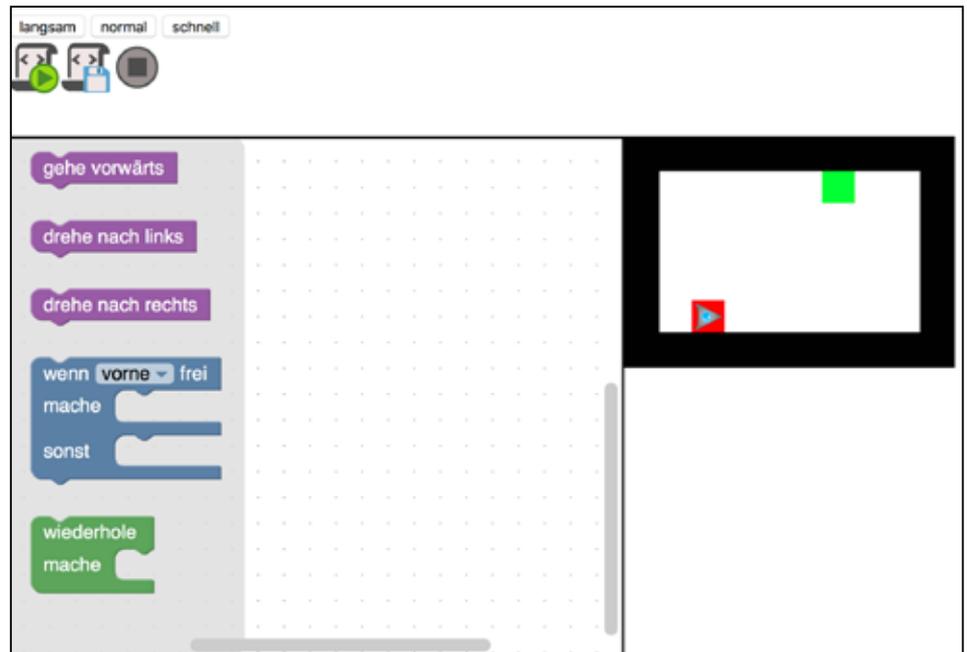
... Programm testen



... Programm testen
und speichern



... Programm
anhalten/
zurücksetzen



So ist es richtig:

Startfeld und Zielfeld liegen an der Wand. Wenn der Roboter immer an der Wand entlang geht, erreicht er sicher das Zielfeld. Dazu kann er wiederholt Folgendes tun: Wenn der Weg vorne freie ist, geht er vorwärts; sonst (dann steht er vor einer Wand) dreht er nach links. Dieses Vorgehen kann man auch mit vier Programmbausteinen beschreiben:



Das ist Informatik!

In dem 8 mal 5 Schritte großen Raum führt der Roboter 14 Bewegungen aus, um das Zielfeld zu erreichen: vor – vor – vor – vor – vor – vor – links – vor – vor – vor – vor – links – vor – vor. Das Programm, das den Roboter so steuert, hat aber nur vier Bausteine. Es macht sich eine Wiederholung (in der Informatik auch Schleife genannt) zunutze, um beliebig viele Vorwärtsschritte und Linkswendungen zu vollführen, bis das Ziel erreicht ist.

Außerdem enthält es eine bedingte Anweisung („wenn ... mache ... sonst ...“), um bis zur nächsten Wand zu gehen und dann die Richtung zu ändern.

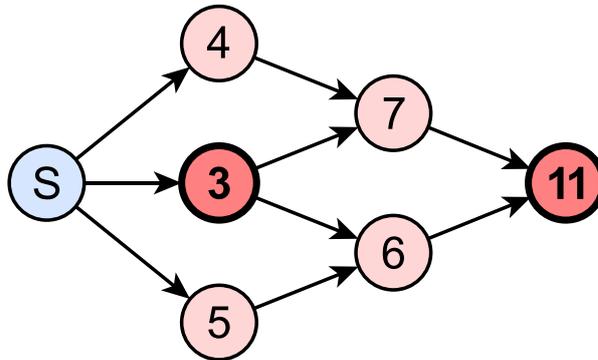
Allerdings findet dieses kurze Programm nicht den kürzesten Weg. Außerdem funktioniert es nicht in jeder Situation. Stelle dir vor, das Ziel läge nicht an einer Wand – würde der Roboter es jemals erreichen?



Kürzeste Wege

Das Bild stellt ein Netzwerk aus Einbahnstraßen (Pfeile) und Kreuzungen (Kreise) dar. Bis auf Kreuzung S ist jede Kreuzung mit einer Zahl markiert. Die Zahl gibt die Länge des kürzesten Weges von S zu dieser Kreuzung an.

Betrachte die beiden mit starkem Rand abgebildeten Kreuzungen.



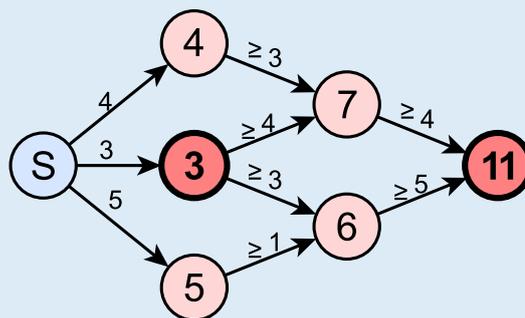
Welche der folgenden Aussagen über diese beiden Kreuzungen ist sicher wahr?

- A) Die Länge des kürzesten Weges zwischen diesen Kreuzungen ist genau 8.
- B) Die Länge des kürzesten Weges zwischen diesen Kreuzungen ist 8 oder weniger.
- C) Die Länge des kürzesten Weges zwischen diesen Kreuzungen ist 8 oder mehr.
- D) Über die Länge des kürzesten Weges zwischen diesen Kreuzungen kann man nichts sagen.

Antwort C ist richtig:

Der kürzeste Weg zwischen diesen beiden Kreuzungen ist 8 oder mehr.

Wäre der kürzeste Weg weniger als 8, so wäre der kürzeste Weg zwischen S und der Kreuzung mit „11“ weniger als $3 + 8 = 11$. Der kürzeste Weg könnte aber länger als 8 sein, da der kürzeste Weg zwischen S und der Kreuzung mit „11“ auch über die Kreuzungen mit „4“ und „7“ oder über die Kreuzungen mit „5“ und „6“ führen könnte. Damit sind die anderen Antworten falsch.



Das Bild zeigt, wie lang die einzelnen Einbahnstraßen mindestens sein müssen, damit die Zahlen der Kreuzungen stimmen.

Das ist Informatik!

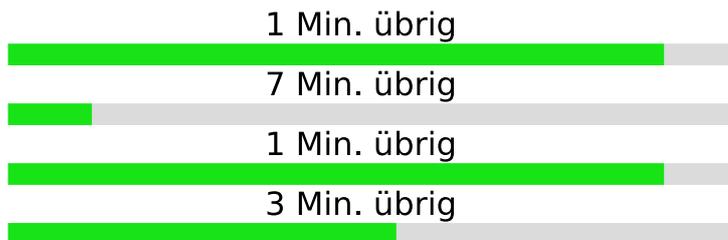
Auf den ersten Blick scheint es in dieser Aufgabe lediglich um den kürzesten Weg zwischen zwei Kreuzungen zu gehen. Dafür müsste man aber die tatsächliche Entfernung zwischen zwei Kreuzungen wissen, die man üblicherweise als Weglänge an den Weg schreibt. In diesem Fall jedoch ist nur die Länge des kürzesten Weges von S zu jeder Kreuzung notiert, es könnte zum Beispiel sein, dass der Weg von „4“ zu „7“ 10 lang ist, wenn der Weg von „3“ zu „7“ 4 lang ist. Daher kann man ohne weitere Informationen nicht sagen, welcher der möglichen Wege der kürzeste Weg ist.



Ladezeit

Lädt man mehrere Dateien gleichzeitig aus dem Internet, dann teilen sich diese Downloads die Kapazität der Internet-Verbindung gleichmäßig. Beim Laden von 10 Dateien gleichzeitig kann jede Datei nur ein Zehntel der Kapazität nutzen; das Laden einer Datei hätte die 10-fache Ladegeschwindigkeit.

Ein Benutzer lädt gerade 4 Dateien gleichzeitig herunter. Für jede Datei wird die verbleibende Ladezeit angezeigt. Diese Zeit wird anhand der aktuellen Verbindungskapazität berechnet:

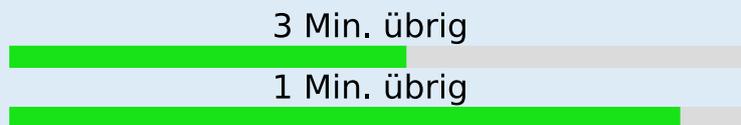


In der Folgezeit bleibt die Verbindungskapazität gleich.

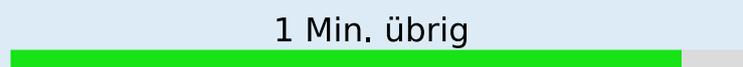
Wie viele Minuten wird es dauern, bis alle 4 Dateien vollständig geladen sind?

3 ist die richtige Antwort:

Nach einer Minute sind zwei der vier Dateien fertig geladen. Blicke die Ladegeschwindigkeit für die verbleibenden einzelnen Dateien gleich, würde deren Download noch 6 und 2 Minuten dauern. Die Ladegeschwindigkeit verdoppelt sich aber, weil die verbleibenden zwei Dateien jetzt die doppelte Kapazität nutzen können. Es verbleiben also 3 Minuten und 1 Minute:



Nach einer weiteren Minute ist die dritte Datei fertig geladen. Die letzte Datei hätte jetzt noch 2 Minuten, aber die Ladegeschwindigkeit verdoppelt sich wieder. Daher verbleibt genau 1 Minute:



Nach insgesamt 3 Minuten sind alle 4 Dateien geladen.

Das ist Informatik!

Die Datenübertragungsrate, auch als Übertragungsgeschwindigkeit oder – nicht ganz zutreffend – als „Bandbreite“ bezeichnet, ist die digitale Datenmenge, die innerhalb einer Zeitspanne über einen Übertragungskanal übertragen wird. Die maximal mögliche Datenübertragungsrate für einen Kanal wird als Kanalkapazität bezeichnet. Für einen Übertragungskanal ist aber auch die sogenannte Antwortverzögerung oder Latenzzeit ein wichtiges Maß. Damit wird die Zeitdauer angegeben, wie lange die Daten vom Sender zum Empfänger brauchen (manchmal wird auch der Rückweg zum Sender miteinberechnet).

Die Kapazität eines Kanals ist fest. Eine einzelne Datenübertragung durch einen Kanal kann diese Kapazität voll ausnutzen. Wenn aber mehrere Datenübertragungen gleichzeitig über den selben Kanal gehen, wird die Kapazität geteilt und die einzelnen Übertragungsraten sinken. Wie lange ein Download aus dem Internet dauert, hängt also nicht nur von der Kapazität der Leitung ab, sondern insbesondere davon, mit wie vielen anderen Übertragungen die Leitung geteilt wird. Ein wichtiges Prinzip im Internet ist, dass die Kapazität eines Kanals gleichmäßig auf die darüber stattfindenden Datenübertragungen verteilt wird; das nennt man „Netzneutralität“. Manche Anbieter von Diensten, die besonders viel Kapazität benötigen, mögen dieses Prinzip nicht. Sie würden gerne etwas dafür bezahlen, dass ihre Daten schneller übertragen werden als andere.



3-4: schwer

5-6: mittel

7-8: –

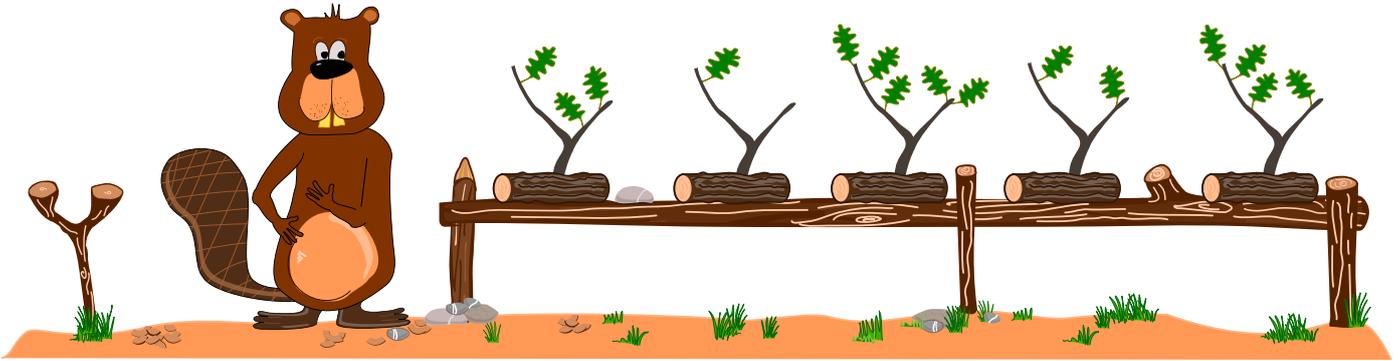
9-10: –

11-13: –



Lecker!

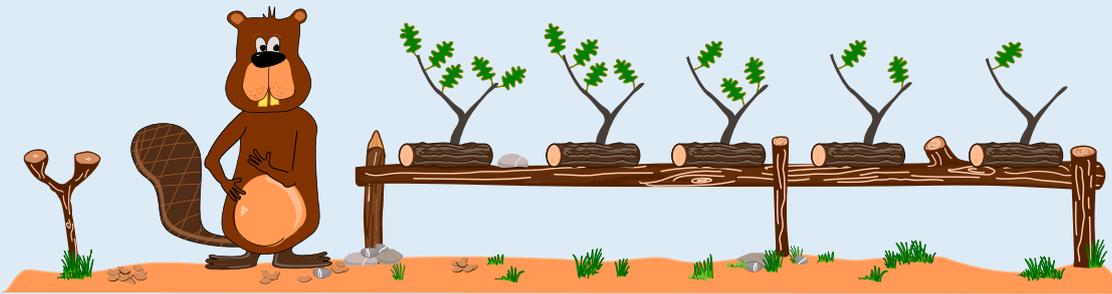
Biber Ben hat auf einem Brett fünf Stämme gesammelt. An den Stämmen wachsen Zweige mit Blättern. Ben liebt Blätter: je mehr Blätter, desto leckerer! Ben möchte, dass die Stämme auf dem Brett anders liegen: je leckerer, desto näher bei ihm. Der Stamm mit den meisten Blättern soll direkt bei ihm liegen.



Lege die Stämme so auf das Brett, wie Ben das möchte!

Bewege dazu die Stämme mit der Maus. Du kannst einen Stamm auf die Ablage neben Ben legen oder auf einen leeren Platz auf dem Brett.

So ist es richtig:

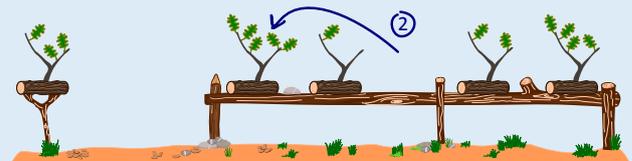


Sortieraufgaben lassen sich immer mit Vertauschungen erledigen. Allerdings kann man sich viele verschiedene Vorgehensweisen überlegen, welche Dinge wann und wie vertauscht werden. Da immer nur ein Stamm bewegt werden kann, wird zum Vertauschen eine Ablage benötigt. Damit kann man z.B. so vorgehen:

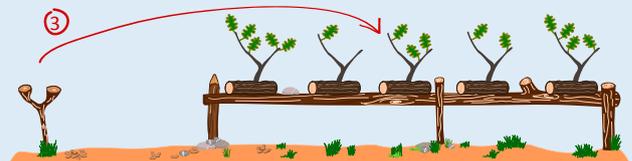
- Bewege den Stamm vom ersten Platz auf die Ablage.



- Bewege den "besten" Stamm (also den mit den meisten Blättern) auf den – frei gewordenen – ersten Platz.



- Bewege den Stamm von der Ablage auf den nun freien Platz.



Damit ist der erste Platz nun mit dem richtigen Stamm belegt. Auf die gleiche Weise werden der zweite Platz mit dem richtigen Stamm belegt, der dritte Platz mit dem richtigen Stamm belegt usw.

Das ist Informatik!

Sortieren ist eine der Hauptbeschäftigungen von Computern. Sie sortieren Daten ganz nach Wunsch: Zahlen nach Größe, Farben nach Helligkeit, Fotos nach dem Zeitpunkt der Aufnahme usw. Damit eine Menge von Daten umsortiert werden kann, müssen die einzelnen Daten ihren „Platz“ tauschen: eine kleinere Zahl muss vor eine größere gesetzt werden, eine hellere Farbe vor eine dunklere, ein älteres Foto vor ein neueres.

Die „Plätze“ im Computer werden in Programmen mit so genannten Variablen adressiert. Wenn man den Wert zweier Variablen tauschen möchte, benötigt man eine dritte Variable, um einen Wert zwischendurch abzulegen – als Tauschhilfe, sozusagen. Denn wenn man ohne nachzudenken den Platz 1 mit dem Wert von Platz 2 belegt, ist der Wert von Platz 1 verschwunden. Die Vertauschung zweier Daten an den Plätzen (Variablen) a und b mit Tauschhilfsplatz t kann so beschrieben werden:

- t ← a (Die Hilfsvariable t wird mit dem Wert der Variable a belegt.)
- a ← b (Die Variable a wird mit dem Wert der Variable b belegt.)
- b ← t (Die Variable b wird mit dem Wert der Hilfsvariable t belegt.)

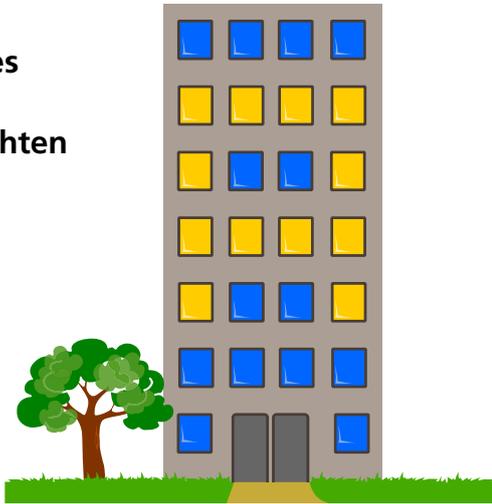
Die Informatik kennt viele Verfahren zum Sortieren von Daten. Wie gut sie funktionieren, wird u.a. an der Anzahl der Vertauschungen gemessen, die zum Sortieren benötigt werden. Bei der Erklärung der Lösung wird übrigens das Verfahren Selection Sort angewendet.



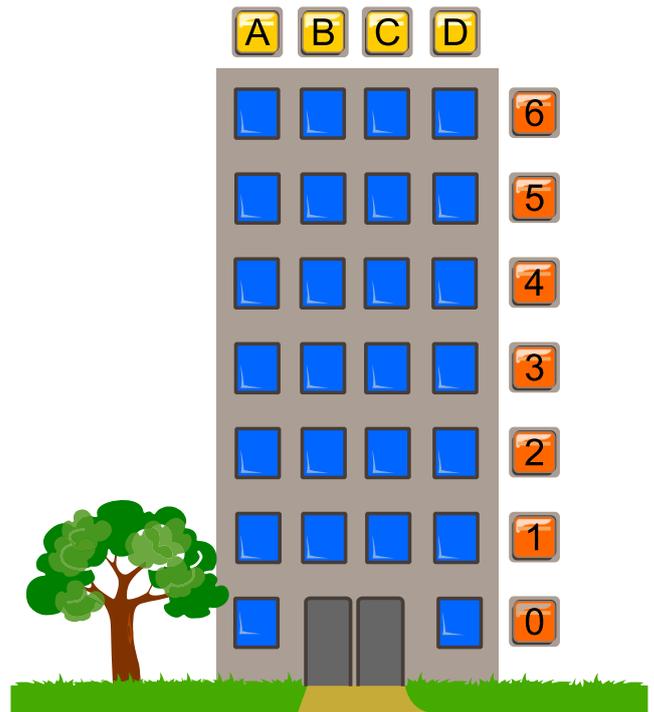
Lichtkunst

Das neue Hochhaus der Stadt hat 26 Fenster, hinter denen das Licht ein- und ausgeschaltet werden kann. Leider kann man das Licht jedes Fensters nicht einzeln ein- und ausschalten, sondern muss immer alle Lichter eines Stockwerks oder einer Fensterspalte vollständig an- oder ausschalten.

Die Lichter des Hochhauses sollen so leuchten wie im Bild.



Klicke auf die Nummern der Stockwerke bzw. Buchstaben der Fensterspalten, um deren Lichter zu schalten.



So ist es richtig:

Die Lichter leuchten so wie im Bild, wenn man zunächst die Stockwerke 3 und 5 anschaltet, danach die Fensterspalten A und D anschaltet und zuletzt die Stockwerke 6, 1 und 0 wieder ausschaltet. Natürlich gibt es noch viele andere Möglichkeiten, die im Kern aber auf dieselbe Abfolge zurückgehen.

Das ist Informatik!

Die Lichtschalter in dieser Biberaufgabe sind wie Anweisungen in einem Computerprogramm, und die Lichter in den Fenstern sind wie Speicherplätze im Computer, die durch diese Anweisungen verändert werden. Dabei teilen sich die Anweisungen (also die Schalter) einige Speicherplätze (also die Lichter): Zum Beispiel wird das Licht im Fenster links oben sowohl vom Schalter für Fensterspalte A als auch vom Schalter für Stockwerk 6 geschaltet.

Wenn in einem Computerprogramm verschiedene Anweisungen die gleichen Speicherplätze verändern, ist es besonders wichtig, dass sie in der richtigen Reihenfolge ausgeführt werden. Sonst entstehen leicht Fehler. Werden z.B. die Lichter in Fensterspalte A eingeschaltet und wieder ausgeschaltet und danach die Lichter in Stockwerk 6 eingeschaltet, ist das Licht im Fenster links oben am Ende an. Wird Stockwerk 6 aber zwischen dem Ein- und Ausschalten von Spalte A angeschaltet, ist dieses Licht am Ende aus. Programmierinnen und Programmierer müssen die Auswirkungen von Programmanweisungen sehr genau bedenken.



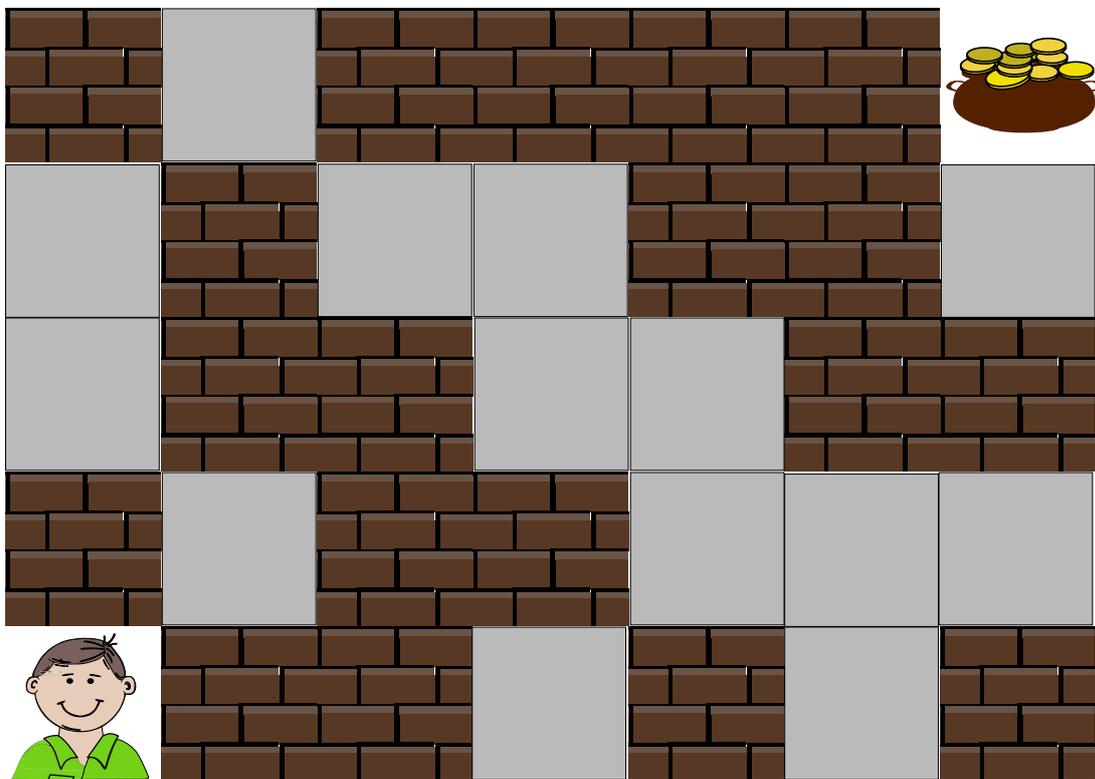
Mauern öffnen

Eine Schatzkarte ist in Felder eingeteilt. Peter ist im Feld links unten und will zum Schatz rechts oben. Er kann immer zum nächsten Nachbarfeld gehen: nach oben, unten, rechts und links – aber nicht schräg.

Peter muss alle Mauern auf seinem Weg zum Schatz öffnen. Das ist anstrengend! Er sucht deshalb einen Weg, auf dem er möglichst wenige Mauern öffnen muss.

Finde diesen Weg!

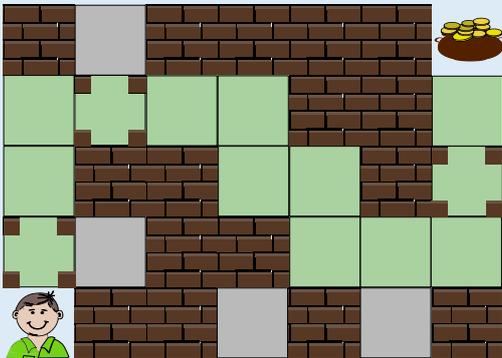
Entferne möglichst wenige Mauern, so dass ein Weg zum Schatz frei wird. Klicke auf ein Mauerfeld, um die Mauer zu öffnen. Klicke erneut auf das Mauerfeld, um die Mauer wieder zu schließen.





So ist es richtig:

Es müssen mindestens drei Mauern entfernt werden, damit für Peter ein Weg zum Schatz frei wird:

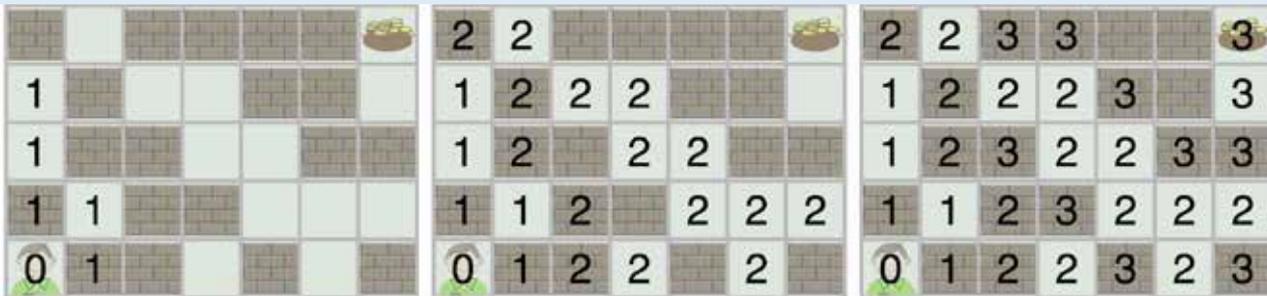


Systematisch kann man das Ergebnis erhalten, wenn man jedes Feld markiert mit der "kleinsten Anzahl an Mauern, die entfernt werden müssen, damit Peter zu diesem Feld kommt".

Wir beginnen mit dem Startfeld links unten und markieren es mit 0 – denn Peter ist schon auf diesem Feld. Weitere freie Felder, die man von dort direkt erreichen kann und deshalb auch mit 0 markiert werden müssten, gibt es nicht.

Dann markieren wir die auf 0 angrenzenden Mauern mit 1, sowie alle freien Felder, die man durch Entfernen einer dieser Mauern nun neu erreichen kann, auch mit 1. Von diesen „1er-Feldern“ aus werden die durch Entfernen irgendeiner weiteren Mauer mit 2 markiert usw. Wenn schließlich das Zielfeld markiert ist, kennt man die kleinste Anzahl an Mauern, die entfernt werden müssen, damit für Peter ein Weg zum Schatz frei wird.

Die Bilder zeigen diesen Vorgang schematisch. Im Zielfeld steht schließlich eine 3.



Das ist Informatik!

Der oben beschriebene Algorithmus zur Lösung dieser Biberaufgabe funktioniert nach einem Prinzip, das in der Informatik als Breitensuche bekannt ist: Man „sucht“ das Zielfeld und betrachtet dazu zuerst alle Nachbarmfelder mit gleicher Anzahl geöffneter Mauern. So geht man, bildlich gesprochen, zuerst in die Breite, bevor man eine weitere Mauer öffnet, sich dadurch etwas weiter vom Startfeld entfernt und damit in die Tiefe des Suchraums abtaucht. Weil immer die gesamte Breite ausprobiert wird, bevor man die Entfernung vergrößert, ist garantiert, dass der kürzeste Weg (gemessen in der Anzahl der zu öffnenden Mauern) zum Ziel gefunden wird. Und weil der Algorithmus die Felder mit der Entfernung vom Startfeld markiert, ist am Ende auch die Länge des kürzesten Weges zum Zielfeld bekannt.

Die Bestimmung kürzester Wege ist – offenkundig – für die Routenplanung eine zentrale Aufgabe. Dabei muss die „Länge“ eines Weges nicht nur durch die Entfernung zwischen Start und Ziel bestimmt sein, sondern kann zusätzlich von weiteren Bedingungen abhängen. Zum Beispiel können Mautgebühren berücksichtigt werden, die auf dem Weg zu zahlen sind – oder Mauern, die auf dem Weg zu öffnen sind.

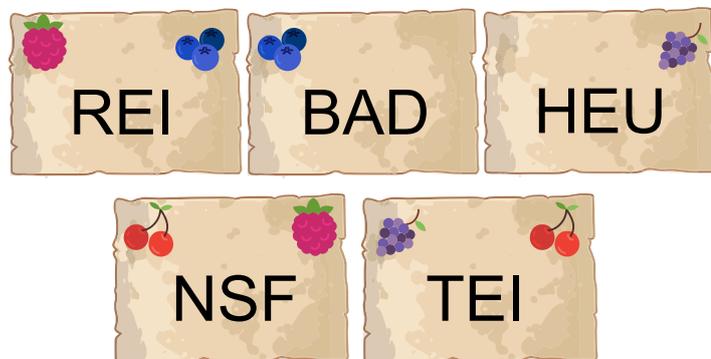


Nachrichtendienst

Viola sendet mit Hilfe der Biber Nachrichten an Leo. Sie teilt eine Nachricht in Stücke mit 3 Buchstaben auf. Jedes Stück schreibt sie auf eine Karte. Dann gibt sie jede Karte einem Biber. Manchmal werden die Biber aufgehalten. Dann kommen die Karten durcheinander bei Leo an. Viola malt deshalb Bilder auf die Karten. So zeigt sie Leo, wie er die Karten lesen muss. Ein Beispiel: Die Nachricht HITZEFREI sendet Viola mit diesen Karten:



Die Biber bringen Leo nun diese Karten:



Welche Nachricht hat Viola gesendet?

- A) BADREIHEUNSFTEI
- B) HEUBADREITEINSF
- C) HEUTEINSFREIBAD
- D) REITEINSHEUBAD

Antwort C ist richtig:

Das Beispiel HITZEFREI hat gezeigt: Leo muss die Karten so legen, dass die gleichen Bilder nebeneinander liegen.



Viola hat die Nachricht HEUTEINSFREIBAD gesendet.

Das ist Informatik!

Wenn Daten wie zum Beispiel E-Mails, Bilder oder Videos im Internet verschickt werden, werden sie in kleine Pakete (TCP/IP-Paket) zerlegt. Jedes Paket hat eine Größe von maximal von 65536 Zeichen ($2^{16} = 65536$ Zeichen = 64 KB).

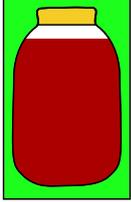
Diese Pakete werden dann teilweise über verschiedene Router mit einigen zusätzlichen Informationen über das Paket selbst (Sender, Empfänger, Sequenznummer des Paketes ...) gesendet. Alle diese Informationen sorgen dafür, dass die Originalinformation, auch wenn die Teilinformationen (Teilpakete) zeitversetzt ankommen, beim Empfänger wieder korrekt zusammengesetzt werden kann.

Anmerkung: Bei IPv6 sind größere Pakete möglich.



Opas Marmelade

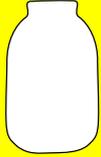
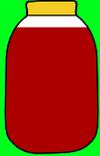
Opa kocht Marmelade. Anna, Peter und Lisa helfen ihm, die Marmelade in Gläser abzufüllen. Dazu müssen sie jeweils diese Arbeitsschritte erledigen – und zwar genau in dieser Reihenfolge:

 <p>Ein Glas spülen. Das dauert 3 Minuten.</p>	 <p>Marmelade in das Glas füllen. Das dauert 2 Minuten.</p>	 <p>Das Glas verschließen. Das dauert 1 Minute.</p>
---	--	--

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 erledigt werden kann.
- Ein Glas muss ganz sauber sein, bevor man Marmelade einfüllt.
- Und das Glas darf natürlich erst verschlossen werden, wenn es fertig gefüllt ist.

Dieser Plan ist also nicht möglich:

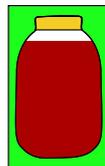
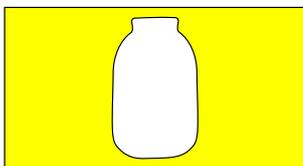
ANNA				
PETER				
LISA				

Anna, Peter und Lisa wollen in 10 Minuten möglichst viele Gläser abfüllen.



Erstelle einen Plan, mit dem sie das schaffen.

ANNA										
PETER										
LISA										



So ist es richtig:

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.

Das Abfüllen eines Marmeladenglases dauert $3 + 2 + 1 = 6$ Minuten. Jedes der drei Kinder hat 10 Minuten Arbeitszeit. Somit beträgt die gesamte Arbeitszeit 30 Minuten. Es ist nicht möglich, in 30 Minuten mehr als fünf Gläser abzufüllen; denn $30 : 6 = 5$. Fünf Gläser sind also auch theoretisch gesehen das Optimum. Es kann mit dem abgebildeten Plan erreicht werden.

ANNA					
PETER					
LISA					

Das ist Informatik!

In dieser Biberaufgabe arbeiten Anna, Peter und Lisa gleichzeitig, also parallel an einer Aufgabe. Die Aufgabe kann zwar glücklicherweise in Teilschritte zerlegt werden aber zwischen den Teilschritten besteht eine Abhängigkeit. Der nächste Teilschritt kann erst begonnen werden, wenn der frühere Teilschritt abgeschlossen ist. Diese sehr häufige Abhängigkeit nennt man „Finish-to-Start“-Abhängigkeit (es gibt auch „Start-to-Start“, „Finish-to-Finish“, „Start-to-Finish“). In einem großen Projekt gibt es sehr viele Teilschritte mit unterschiedlichen Abhängigkeiten. Zur Darstellung der Reihenfolge und Abhängigkeiten solcher Teilschritte werden Netzpläne verwendet. Innerhalb jedes solchen Netzplanes gibt es einen sogenannten „kritischen Pfad“. Er bestimmt die Projektdauer und Verzögerungen in Teilschritten entlang dieses Pfades verlängern das Projekt.

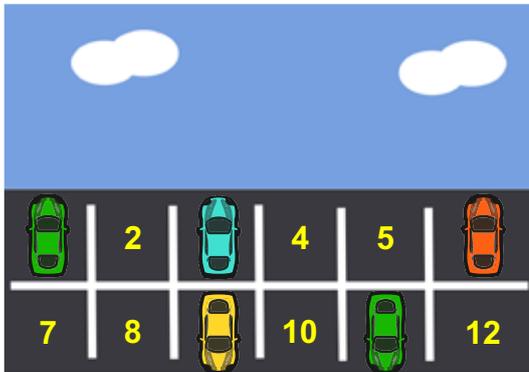
Auch innerhalb eines Computers spielen solche Überlegungen eine wichtige Rolle. Ein Prozessor besteht heute meist aus mehreren Kernen (Recheneinheiten). Diese arbeiten parallel und wenn Aufgaben in Teilschritte zerlegbar sind, dann können diese durch verschiedene Kerne gleichzeitig abgearbeitet werden und somit schneller beendet werden.



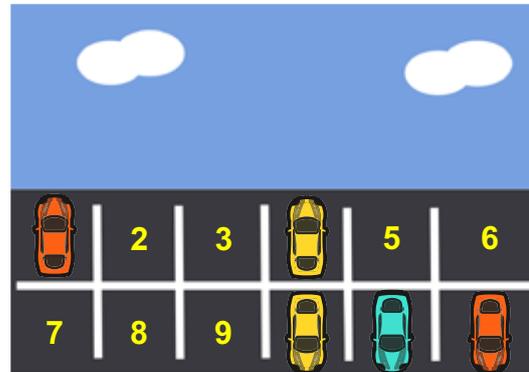
Parkplätze

Ein Geschäft hat 12 Parkplätze für Autos. Die Parkplätze haben Nummern. Die Bilder zeigen, auf welchen Parkplätzen am Montag und am Dienstag Autos waren.

Montag



Dienstag



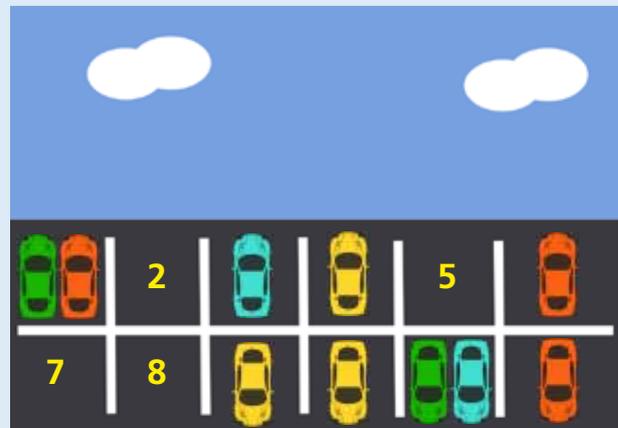
Einige Parkplätze waren an beiden Tagen frei, zum Beispiel Parkplatz 2.

Wie viele Parkplätze waren an beiden Tagen frei?

- A) 2 Parkplätze
- B) 4 Parkplätze
- C) 7 Parkplätze
- D) 14 Parkplätze

Antwort B ist richtig:

Wenn die Autos von beiden Tagen zusammen auf ihren Parkplätzen gezeigt werden, sehen wir, welche Parkplätze an beiden Tagen frei waren. Diese freien Plätze können dann einfach gezählt werden. Insgesamt waren 4 Parkplätze an beiden Tagen frei.



Das ist Informatik!

Alle Arten von Daten können als eine Folge von Nullen und Einsen codiert werden. Eine solche Folge heißt Binärcode. Jede Stelle, wo eine Null oder Eins stehen kann, wird Bit genannt.

Bei dieser Aufgabe können wir die Anwesenheit eines Autos auf einem Parkplatz mit einer Eins (1) und eine Abwesenheit mit einer Null (0) codieren. So lässt sich die Belegung der Parkplätze als Binärcode modellieren: Für Montag ergibt sich der Binärcode 101001001010, für Dienstag 100100000111.

Mit Hilfe der logischen Operation ODER (englisch: OR) wird aus zwei Binärcodes ein neuer so berechnet, dass im Ergebnis alle diejenigen Bits 1 sind, die in mindestens einem der beiden Binärcodes auch 1 sind. Mit ODER erhalten wir also einen Parkplatz-Belegungs-Binärcode, der zeigt, wo an beiden Tagen Autos standen. Das Ergebnis von ODER können wir gut berechnen, indem wir die Binärcodes übereinander schreiben.

$$\begin{array}{r}
 101001001010 \\
 \text{ODER } 100100000111 \\
 = 101101001111
 \end{array}$$

Der Ergebnis-Binärcode enthält 4 Nullen. Es waren also 4 Parkplätze an beiden Tagen frei.



3-4: -

5-6: -

7-8: schwer

9-10: -

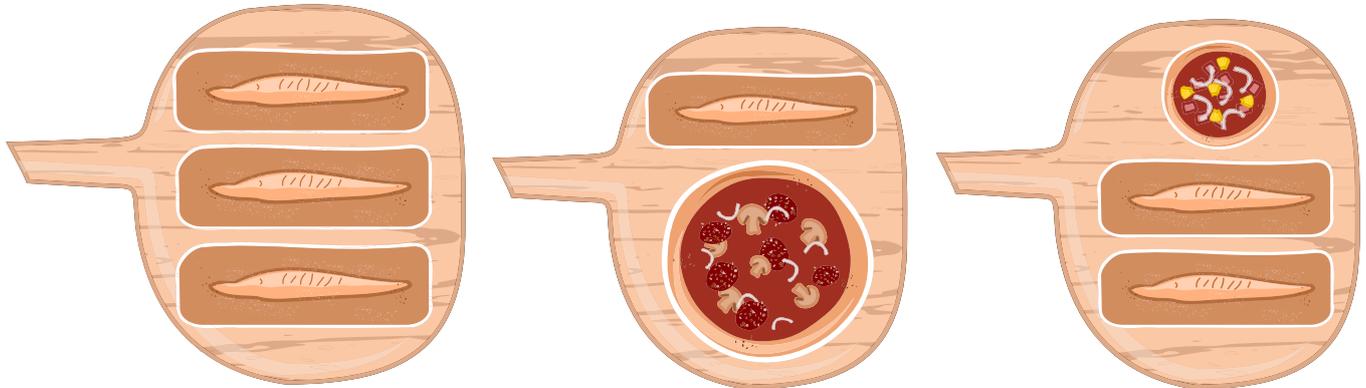
11-13: -



Pizzeria Biberia

Die Pizzeria Biberia hat nur einen kleinen Pizzaofen.

Deshalb kann der Pizzabäcker nur wenige Gerichte gleichzeitig backen, nämlich:



Der Pizzabäcker kann aber Gerichte einzeln in den Ofen legen oder aus dem Ofen holen. Die Gerichte müssen unterschiedlich lange im Ofen sein: eine kleine Pizza 10 Minuten, eine große Pizza 15 Minuten und ein Ciabatta-Brot 20 Minuten.

Heute ist viel los: Die Gäste bestellen

- eine kleine Pizza,
- zwei große Pizzen und
- vier Ciabatta-Brote

1x

2x

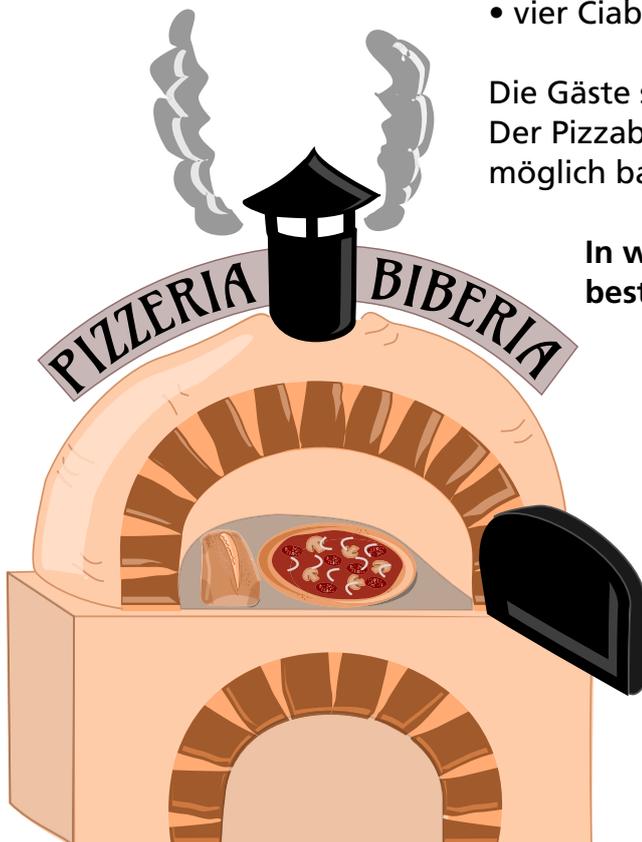
4x



Die Gäste sind hungrig.

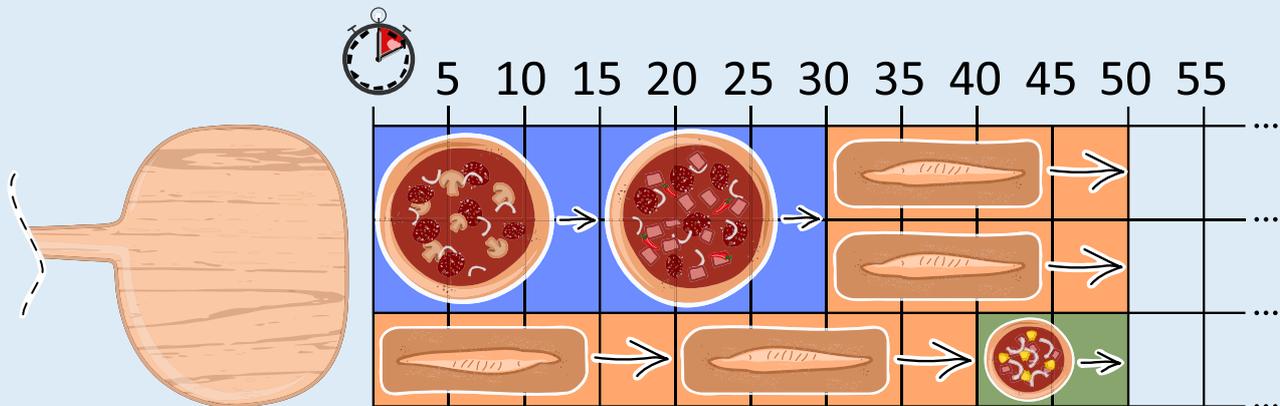
Der Pizzabäcker soll die bestellten Gerichte so schnell wie möglich backen.

In wie vielen Minuten kann der Pizzabäcker die bestellten Gerichte so schnell wie möglich backen?



**50 ist die richtige Antwort:**

Der Pizzabäcker kann die beiden großen Pizzen hintereinander backen, parallel dazu zwei Ciabatta-Brote und dann mit zwei Ciabatta-Broten und einer kleinen Pizza ergänzen:



Der Pizzabäcker kann die Gerichte also in 50 Minuten backen. Es gibt noch weitere Möglichkeiten, die Gerichte in 50 Minuten zu backen; z.B. kann der Pizzabäcker den abgebildeten Plan in umgekehrter Reihenfolge ausführen. Schneller kann es nicht gehen, da der Platz im Ofen während der 50 Minuten immer komplett ausgenutzt wird.

Das ist Informatik!

Wenn man versucht, einen Ablaufplan (engl. schedule) für eine beschränkte Ressource (in diesem Fall der Pizzaofen) zu erstellen, möchte man in der Regel eine optimale Lösung finden, beispielsweise bezüglich der Kosten oder (wie in diesem Fall) bezüglich der Zeit. So wird die Ressource optimal genutzt und damit auch die kürzeste Backzeit erreicht.

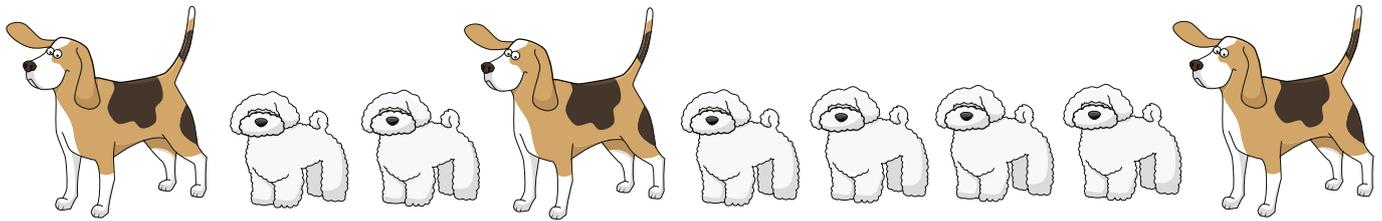
Ein weit verbreitetes Verfahren zum Finden von optimalen Ablaufplänen ist das Rundlauf-Verfahren (engl. Round-Robin). Bei diesem Verfahren sind alle Prozesse, die stattfinden sollen, in einer Warteschlange. Jeder Prozess bekommt dann für eine kurze Zeit die Ressource zur Verfügung gestellt. Wenn er am Ende der kurzen Zeit noch nicht fertig ist, wird er wieder neu in die Warteschlange eingereiht. Für diesen Pizzaofen jedoch macht die Strategie nicht viel Sinn ... man nimmt ja keine halbfertige Pizza aus dem Ofen und schiebt sie später wieder rein. Daher kann man in diesem Fall mit einer gierigen (engl. greedy) Strategie vorgehen: man fängt mit dem größten Prozess an (einer großen Pizza, da diese am meisten Platz wegnimmt) und versucht dann, mit nächstkleineren Prozessen aufzufüllen und so weiter. Da der Ofen (mit Einschränkungen) mehrere Dinge gleichzeitig backen kann, kann man auch immer jeweils das fertige Gericht herausnehmen und durch das nächste zu backende Gericht ersetzen.

Hat man aber am Ende die optimale Lösung gefunden? Das ist nicht offensichtlich, man könnte ja vielleicht eine bessere Lösung finden, indem man beispielsweise mit drei Ciabatta-Broten anfängt. Wenn man es aber geschafft hat, alle Ressourcen immer voll zu belegen (also in diesem Fall den Ofen immer voll zu haben), kann man sicher sein, dass es keine bessere Lösung gibt.

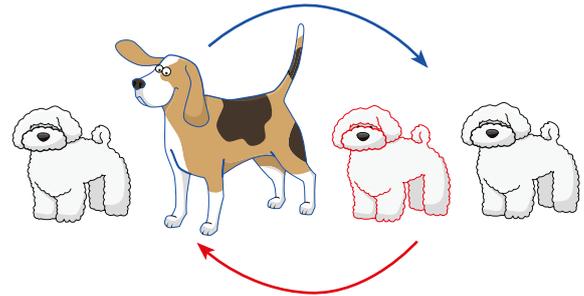


Platz! Tausch!

Hier ist eine Reihe von Hunden. Es gibt große und kleine Hunde.



Zwei Hunde, die nebeneinander stehen, können einen Platztausch machen: Sie tauschen dann ihre Plätze in der Reihe. Die großen Hunde sollen nebeneinander stehen.



Wie viele Platztausche müssen die Hunde dazu mindestens machen?

- A) 2 Platztausche B) 6 Platztausche
C) 8 Platztausche D) 10 Platztausche

Antwort B ist richtig:

Der große Hund links macht zwei Platztausche mit den zwei kleinen Hunden rechts neben ihm. Dann steht er neben dem großen Hund in der Mitte. Der große Hund rechts macht vier Platztausche mit den vier kleinen Hunden links neben ihm. Dann steht auch er neben dem großen Hund in der Mitte. Insgesamt sind das sechs Platztausche.

Alle kleinen Hunde stehen zwischen großen Hunden. Wenn die großen Hunde nebeneinander stehen sollen, muss also mindestens jeder kleine Hund mit einem großen Hund einen Platztausch machen. Sechs Platztausche müssen die Hunde also mindestens machen.

Wenn die großen Hunde links nebeneinander stehen sollen, müssen die Hunde acht Platztausche machen. Wenn die großen Hunde rechts nebeneinander stehen sollen, müssen die Hunde zehn Platztausche machen.

Das ist Informatik!

Daten werden im Memory eines Computers gespeichert. Dies beinhaltet sowohl den internen Speicher (auch als RAM bezeichnet) und den externen Speicher (z.B. eine Festplatte, ein USB-Stick). Computer können im internen Speicher schneller auf Daten zugreifen, jedoch der externe Speicher ist kostengünstiger als der interne Speicher. Das bedeutet, dass moderne Computer in der Regel mehr externen Speicher haben. Wissenschaftler suchen nach Wegen, um den internen Speicher so oft wie möglich zu nutzen und den externen Speicher so wenig wie möglich.

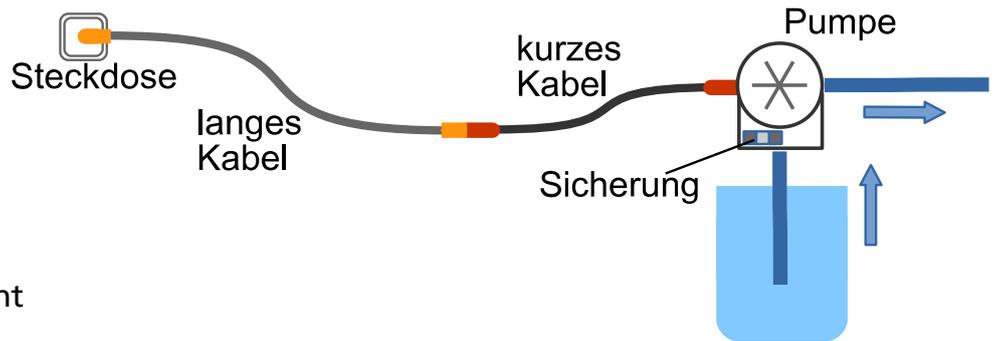
Bei dieser Biberaufgabe geht es im Wesentlichen um Vertauschungen von jeweils zwei Elementen. Wenn wir die Hunde als Daten im Speicher betrachten, dann beinhaltet eine Vertauschung die Änderung der Lage von zwei Dateneinheiten im Speicher. Sind diese Daten im externen Speicher, so gilt es, möglichst wenige Vertauschungen für die Lösung der Aufgabe ausführen zu müssen.

Vertauschungen kommen auch bei verschiedenen Sortieralgorithmen vor. Auch hier gilt, dass ein effizientes Sortierverfahren möglichst wenig Vertauschungen benötigt. Zum Beispiel werden beim Bubblesort-Verfahren in jedem Durchgang zwei direkte Nachbarn vertauscht, falls diese nicht in der gewünschten Reihenfolge sind.



Pumpensystem

Ein Biber hat für seinen Gemüsegarten ein elektrisches Pumpensystem gebaut. Es ist über eine Steckdose mit dem Stromnetz verbunden und besteht aus diesen Teilen:



- einem langen Kabel
- einem kurzen Kabel
- einer Sicherung und
- einer Pumpe.

Eines Tages funktioniert das Pumpensystem nicht mehr. Der Biber nimmt an, dass nur ein Teil defekt ist. Das möchte er finden, indem er Teile austauscht. Er hat nämlich gebrauchte, aber garantiert voll funktionstüchtige Ersatzteile.

Von seinen Freunden erhält der Biber verschiedene Empfehlungen, wie er vorgehen soll:

- A) Der Biber muss zuerst die Pumpe austauschen, denn sie ist das wichtigste Teil.
- B) Der Biber soll die Teile nach und nach einzeln durch die Ersatzteile austauschen. Wenn das System nach dem Austausch eines Teils wieder funktioniert, dann war dieses Teil defekt.
- C) Der Biber sollte lieber neue Teile besorgen. Die Ersatzteile sind ja schon gebraucht.
- D) Der Biber sollte nach und nach immer zwei Teile gleichzeitig austauschen. So kommt er schnell zum Ziel.

Wie soll der Biber vorgehen?

Wähle die Empfehlung, mit der der Biber ein einzelnes defektes Teil sicher finden wird.

Antwort B ist richtig:

- A) Diese Empfehlung betrifft nur die Pumpe und hilft nicht weiter, wenn ein anderes Teil defekt ist. Außerdem: Wie wichtig ein Teil ist, sagt nichts über seine Fehlerwahrscheinlichkeit aus.
- B) Dieser Test ist sicher, denn so prüft man genau alle verdächtigen Situationen die zur Annahme „genau ein Teil ist defekt“ passen.
- C) So lässt sich zwar das Pumpensystem wieder ans Laufen bringen, aber ein einzelnes defektes Teil wird nicht gefunden. Außerdem: Auch ganz neue Teile sind manchmal defekt.
- D) Wenn der Biber nach dieser Empfehlung vorgeht und immer zwei Teile tauscht, die noch nicht getauscht wurden, kann er ein einzelnes defektes Teil nicht sicher finden. Das klappt nur – beim Tausch von zwei Teilen – wenn er ein bereits ausgetauschtes Teil noch einmal austauschen kann. Es ist aber nicht garantiert, dass es für die einzelnen Teile jeweils mehr als ein Ersatzteil gibt.

Das ist Informatik!

Das Suchen nach Fehlern ist für die Informatik eine fortwährende Aufgabe. Die Hardwarekomponenten und Softwaremodule von Informatiksystemen sind meist komplex. Und die Benutzer der Systeme sind verblüffend kreativ im Produzieren irregulärer Zustände. Es gibt viele Arten von Fehlern in allen möglichen Phasen des Entwurfs, der Realisierung und des Betriebs. Die wichtigste Methode der Fehlersuche ist das systematische Testen. Damit sollte man schon in der Entwurfsphase eines Informatiksystems beginnen und dann niemals aufhören. Auch im laufenden Betrieb können jederzeit neue Fehler entstehen, wenn sich zum Beispiel in der Systemumgebung etwas ändert. Sehr schwer sind Zustände aufzulösen, bei denen mehrere Fehler interferieren. Zum Beispiel würde der in dieser Biberaufgabe empfohlene Test nicht helfen, wenn entgegen der Annahme unseres optimistischen Gärtners doch zwei Teile defekt sind. Testen ist teuer! Hier steckt die Informatik in einer wirtschafts-ethischen Klemme: Wann darf sie annehmen, dass ausreichend getestet wurde?



Punkte sammeln

2	0	1	1	Z
1	2	0	2	3
2	2	0	2	1
3	1	0	2	0
S	0	1	3	0

In die Felder einer Tabelle sind Zahlen eingetragen. Außerdem gibt es ein Startfeld S und ein Zielfeld Z. Die Aufgabe lautet: Finde einen Weg vom Start zum Ziel, auf dem insgesamt maximal viele Punkte gesammelt werden. In jedem Schritt darfst du nur um ein Feld nach rechts oder nach oben gehen, wie die Pfeile zeigen. Die Zahl in jedem Feld gibt an, wie viele Punkte du dort sammeln kannst. Die Punkte aller Felder eines Weges werden beim Sammeln addiert.

Wie viele Punkte können in der Tabelle oben maximal gesammelt werden?

14 ist die richtige Antwort:

Es können maximal 14 Punkte gesammelt werden. Das lässt sich mit der folgenden Hilfstabelle nachweisen. Die Hilfstabelle soll in jedem Feld die maximal erreichbare Punktzahl enthalten für den Fall, dass dieses Feld das Zielfeld wäre. Die Zahl im Feld rechts oben, dem eigentlichen Zielfeld, ist dann das richtige Ergebnis.

Man erhält diese Hilfstabelle so: An der Stelle des Startfeldes trägt man eine 0 ein; denn wenn das Startfeld auch das Zielfeld wäre, könnte man genau 0 Punkte sammeln. Für jedes andere Feld ermittelt man das Maximum der beiden Zahlen, die im links davon und darunter liegenden Feld (jeweils: falls vorhanden) eingetragen sind. Nur von diesen beiden Feldern aus kann man in einem Schritt zum aktuellen Feld gelangen. Zu diesem Maximum addiert man die Punktzahl, die man in der Spiel-tabelle für das aktuelle Feld sammeln kann. Damit enthält auch das aktuelle Feld die maximale Punktzahl, die auf einem Weg vom Start zum aktuellen Feld gesammelt werden kann.

Hier ist die Hilfstabelle. Der Weg, auf dem 14 Punkte gesammelt werden können, ist grau hinterlegt.

8	9	10	12	14
6	9	9	11	14
5	7	7	9	10
3	4	4	6	6
0	0	1	4	4

Das ist Informatik!

Das menschliche Gehirn ist sehr geschickt und effizient im Lösen von anschaulichen Aufgaben. So mancher sieht die Lösung dieser Biberaufgabe nahezu auf einen Blick, eventuell nach kurzem Prüfen weniger Alternativen. Diese intuitive Methode arbeitet aber nicht fehlerfrei und funktioniert nur für Tabellen von überschaubarer Größe. Die oben beschriebene Methode hingegen funktioniert, falls die Einzelschritte richtig ausgeführt werden, immer. Sie kann auch von einem Programm ausgeführt werden und ist auf Tabellen beliebiger Größe anwendbar. Die letztere Eigenschaft nennt man *Skalierbarkeit* und ist ein Kriterium für einen guten Algorithmus.

Das beschriebene Verfahren mit der Hilfstabelle arbeitet nach dem Prinzip der *dynamischen Programmierung*. Es kann angewandt werden, wenn ein Problem (hier: Finde den optimalen Weg!) aus Teilproblemen besteht (hier: Finde einen optimalen Teilweg!) und sich die Lösung des gesamten Problems aus Lösungen von Teilproblemen berechnen lassen kann. Dann berechnet man zunächst die Lösungen der kleinsten Teilprobleme und dann nach und nach die Lösungen größerer Teilprobleme aus den Lösungen passender kleinerer Teilprobleme. Die Lösung eines Teilproblems wird gespeichert und muss deshalb nur einmal berechnet werden.



Saftladen

Vier Freunde halten bei einem Laden. Dort möchte jeder ein Glas Saft trinken. Die Tabelle zeigt, welchen Saft jeder wie gerne trinkt: je mehr Herzen, desto lieber. Der Saftladen ist sehr beliebt und hat von jedem Saft nur noch je ein Glas übrig. Jeder muss also einen anderen Saft bestellen. Die Freunde wollen so bestellen, dass sie insgesamt möglichst zufrieden sind – also möglichst viele Herzen sammeln.

Bestelle für die vier Freunde so, dass sie insgesamt möglichst viele Herzen sammeln!
Klicke jeweils ein Glas in der Tabelle an, um für eine Person ein Glas Saft zu bestellen.

Anna				
Bernard				
Christine				
Daniel				



So ist es richtig:

Anna				
Bernard				
Christine				
Daniel				

Mit dieser Bestellung können die Freunde insgesamt 14 Herzen sammeln.

Um auf diese Lösung zu kommen, kann man von Daniel ausgehen. Für ihn bestellt man den Saft , den er mit vier Herzen mag, alle anderen aber nur mit einem Herz mögen.

Wenn man dann für Bert oder Clara den Saft  bestellt, kann man für die beiden übrigen (Anna und Clara bzw. Anna und Bert) jeweils ihren zweitliebsten Saft bestellen.

Drei der vier Freunde mögen am liebsten den Saft . Da aber nur ein Glas davon vorhanden ist, müssen zwei sich mit ihrem zweitliebsten Saft zufrieden geben. Es kann also keine Bestellung mit mehr als $3 + 3 + 4 + 4 = 14$ Herzen geben.

Es gibt auch nur diese beiden Bestellmöglichkeiten mit 14 Herzen, da alle anderen Möglichkeiten von mindestens einem der Freunde verlangen, dass er seinen drittliebsten Saft mit 2 Herzen trinkt, so dass maximal $2 + 3 + 4 + 4 = 13$ Herzen gesammelt werden können.

Das ist informatik!

In dieser Biberaufgabe geht es darum, die Anzahl der Herzen und damit die Zufriedenheit der vier Freunde zu optimieren. Optimierung ist ein Forschungsschwerpunkt der Informatik und auch der Mathematik. Optimierung wird überall gebraucht.

Optimierungs-Algorithmen rechnen oft sehr lange. Bei unseren vier wählerischen Urlaubern müsste ein nicht spezieller Algorithmus schon ca. 65000 mögliche und unmögliche Fälle der Reihe nach durchprobieren. Durch geschicktes Überlegen kann man diesen Aufwand zwar drastisch reduzieren. Es gibt ja nur 24 mögliche Lösungen, für welche die Anzahl der Herzen ausgerechnet werden muss. Doch solches Überlegen erfordert Optimierungs-Spezialisten.

Das Problem dieser Biberaufgabe ist zum Beispiel eine Sonderform des Matching-Problems. Jeder der vier Personen soll genau ein Getränk zugewiesen werden und es gibt von jedem der vier Getränke nur genau eines. Zudem soll die gemeinsame Zufriedenheit möglichst hoch sein. Optimierungs-Probleme mit dieser Struktur kommen im alltäglichen Leben oft vor. Achte mal darauf!



Schülerzeitung

Bei der Schülerzeitung arbeiten zehn Freiwillige mit.

Immer freitags benutzen sie ihre Freistunden dazu.

Der Stundenplan zeigt farblich an, wer wann Zeit hat und arbeiten will.

Uhr Frei- willige	8	9	10	11	12	13	14	15
1		■	■					
2			■	■	■	■		
3	■	■						
4					■	■	■	
5		■	■					
6				■	■			
7			■	■		■	■	
8		■						
9	■	■	■					
10						■	■	

Für ihre Arbeit soll die Schülerzeitung neue Notebooks bekommen.

Zu jeder Zeit soll jede(r) Freiwillige mit einem eigenen Notebook arbeiten können.

Wenn der Stundenplan gleich bleibt, wie viele Notebooks muss die Schülerzeitung dann **mindestens** bekommen?

- A) vier Notebooks
- B) fünf Notebooks
- C) sieben Notebooks
- D) zehn Notebooks

**Antwort B ist richtig:**

Antwort A ist falsch. Sie wäre passend, wenn zum Beispiel nach der maximalen Anzahl von Arbeitsstunden der einzelnen Freiwilligen gefragt worden wäre. Wurde es aber nicht.

Antwort B ist richtig. Wenn eine bestimmte Anzahl von Freiwilligen gleichzeitig eine Freistunde haben, brauchen sie auch genau die gleiche Anzahl Notebooks. In den Stunden 9 Uhr bis 10 Uhr sowie 10 Uhr bis 11 Uhr wollen jeweils fünf Freiwillige gleichzeitig arbeiten, in allen anderen Stunden weniger. Es genügt also, wenn die Schülerzeitung fünf Notebooks bekommt. Mit denen können die Freiwilligen so arbeiten wie im Bild gezeigt.

Freiwillige \ Uhr	8	9	10	11	12	13	14	15
1		NB 3	NB 3					
2			NB 1	NB 1	NB 1	NB 1		
3	NB 1	NB 1						
4					NB 3	NB 3	NB 3	
5		NB 4	NB 4					
6				NB 2	NB 2			
7			NB 5	NB 5		NB 5	NB 5	
8		NB 5						
9	NB 2	NB 2	NB 2					
10						NB 2	NB 2	

Die Antworten C und D sind falsch. Sie stehen im Widerspruch zur richtigen Antwort B. Sieben bzw. zehn Notebooks sind jeweils mehr als fünf.

Das ist Informatik!

Will man eine Anzahl von Daten in ihrer Anordnung verstehen, so bietet die Informatik erprobte Darstellungsmethoden. Das beginnt mit Listen gleichartiger Elemente, die – manchmal zufällig, manchmal einer Ordnung folgend – „aufgereiht“, „verkettet“, „verzeigert“ etc. sind. So eine Liste könnte zum Beispiel eine Schulklasse darstellen, mit ihren Schülern als Elementen, mit dem „Alter“ als Ordnung.

Mehr Struktur bieten „Tabellen“ an. Der „Stundenplan“ in dieser Biberaufgabe ist eine Tabelle und besteht aus lauter Elementen der Art „von Uhrzeit bis Uhrzeit kann ein Schüler diesen Computer benutzen.“ Die Spalten der Tabelle sind zeitlich nach Stunden, die Zeilen nach den Schülernummern geordnet. Speichert man den Verlauf eines Freitags in diese Tabelle, dann entsteht eine Tabelle „Protokoll“ mit Elementen der Art „von 10 Uhr bis 11 Uhr hat der Freiwillige 5 das Notebook 4 benutzt“ – wie im Bild oben.

Programmtechnisch kann eine Tabelle Ineffizienz verursachen, wenn sie zu groß angelegt wird und weitgehend leer bleibt. Die Information, die die Tabelle oben repräsentiert – also die Angaben dazu, wer wann an welchem Notebook arbeitet – lässt sich auch kompakter repräsentieren, wie die letzte Tabelle zeigt. Die Informatik sieht den Speicherbedarf als zweites großes Maß für die Qualität ihrer Systeme, neben der Laufzeit.

Freiwillige \ Uhr	8	9	10	11	12	13	14	15
1	Freiw. 3	Freiw. 3	Freiw. 2	Freiw. 2	Freiw. 2	Freiw. 2		
2	Freiw. 9	Freiw. 9	Freiw. 9	Freiw. 6	Freiw. 6	Freiw. 10	Freiw. 10	
3		Freiw. 1	Freiw. 1		Freiw. 4	Freiw. 4	Freiw. 4	
4		Freiw. 5	Freiw. 5					
5		Freiw. 8	Freiw. 7	Freiw. 7	Freiw. 7	Freiw. 7	Freiw. 7	



3-4: –

5-6: schwer

7-8: mittel

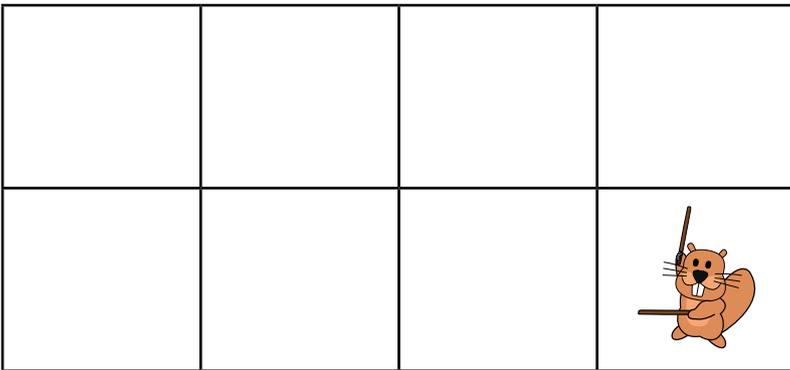
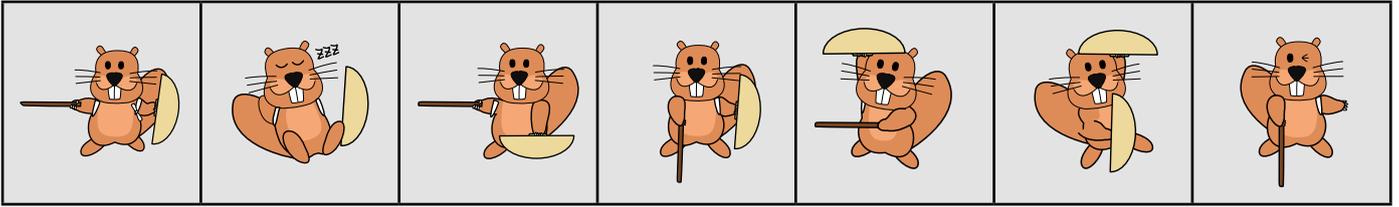
9-10: leicht

11-13: –



Stock und Schild

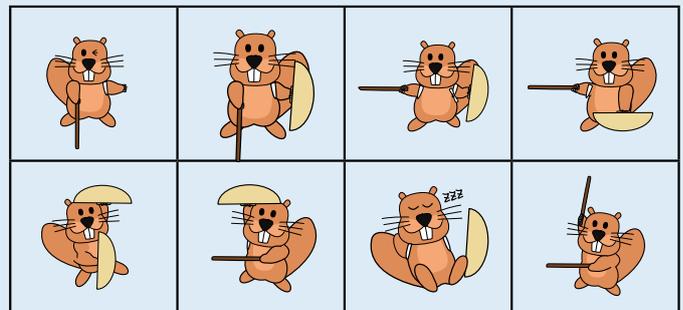
Lucia und ihre Freunde sind Anhänger eines japanischen Spiels mit Stock und Schild. Für ein Foto möchten sie sich auf dem Schulhof so aufstellen, dass jeder Stock auf ein Schild zeigt. Dafür wurden Felder auf den Schulhof gezeichnet. Lucia hat sich bereits in Pose gestellt. Die Bilder darunter zeigen die Freunde in ihren Lieblingsposen.



Schiebe die Bilder der Freunde in die Felder auf dem Schulhof.
Am Ende muss jeder Stock auf ein Schild zeigen.

So ist es richtig:

Die Bilder der Freunde müssen so in die Felder bewegt werden, wie es im Bild unten 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 $1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 = 720$ 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 systematischen Ausprobieren aller Lösungsmöglichkeiten nach dem Prinzip von "Versuch und Irrtum" ist das Backtracking (<https://de.wikipedia.org/wiki/Backtracking>). Ein solches Verfahren ist aber nur dann schnell genug, wenn der Suchraum klein ist. Deshalb ist das Eingrenzen durch logische Regeln so wichtig.

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



Viertel

Schwarz-weiße Pixelbilder können mit den Binärzeichen 0 und 1 so dargestellt werden:
Eine 0 steht für ein weißes Pixel, eine 1 für ein schwarzes Pixel.
Ein Bild mit 4 mal 4 Pixeln wird so mit 16 Zeichen dargestellt.

Viele Bilder kann man mit weniger Zeichen darstellen, wenn man das Verfahren Viertel anwendet.

Dazu werden die Zeichen in einem quadratischen Raster angeordnet.
Das Verfahren Viertel wird auf das Raster so angewendet:

- Falls alle Zeichen im Raster 0 sind, ist das Ergebnis das Zeichen 0. (s. Bild links)
 - Falls alle Zeichen im Raster 1 sind, ist das Ergebnis das Zeichen 1.
- Andernfalls wird das Raster in vier gleich große Teil-Raster aufgeteilt.
Das Verfahren Viertel wird dann der Reihe nach auf diese Teil-Raster angewendet, von links oben aus im Uhrzeigersinn.
Das (Gesamt-)Ergebnis entsteht so:
Die vier (Teil-)Ergebnisse werden hintereinander geschrieben, zwischen die Klammerzeichen (und) (s. Bild mitte und rechts).

```
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
```

0

```
1 1 | 0 0
1 1 | 0 0
-----
1 1 | 1 1
1 1 | 1 1
```

(1 0 1 1)

```
1 1 | 0 0
1 1 | 0 0
-----
1 1 | 0 1
1 1 | 0 1
```

(1 0 (0 1 1 0) 1)

Beachte: Wenn ein Raster aus nur einem Zeichen besteht, ist das Ergebnis genau dieses Zeichen.

```
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 0 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
```

Hier ist das Zeichen-Raster für ein Bild mit 8 mal 8 Pixeln.

**Wende das Verfahren Viertel auf dieses Raster an.
Was ist das Ergebnis?**

- A) (1 1 1 0)
B) (1 1 (1 0 1 1) 1)
C) (1 1 1 (1 (1 1 0 1) 1 1))
D) (1 1 1 (1 (1 0 1 1) 1 1))



Antwort D ist richtig:

$$\begin{array}{cccc|cccc}
 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 \hline
 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\
 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 \hline
 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 \hline
 \underline{\underline{(111(1(1011)11))}}
 \end{array}$$

Die Anwendung des Verfahrens Viertel verläuft so (siehe auch das Bild):

Zunächst sind nicht alle Zeichen im Raster gleich. Das Raster wird also in vier Raster aufgeteilt.

Das Gesamtergebnis wird also so aussehen:

(<Ergebnis 1> <Ergebnis 2> <Ergebnis 3> <Ergebnis 4>)

In den Teil-Rastern 1, 2 und 3 sind jeweils alle Zeichen gleich 1. Für diese Raster ist das Ergebnis deshalb

1. Das Gesamtergebnis wird also so aussehen:

(1 1 1 <Ergebnis 4>)

Das Teil-Raster 4 muss aufgeteilt werden. In den Teil-Rastern 4.1, 4.3 und 4.4 sind jeweils wieder alle Zeichen gleich 1. Damit wissen wir, dass das Gesamtergebnis so aussehen wird:

(1 1 1 (1 <Ergebnis 4.2> 1 1))

Teil-Raster 4.2 muss aufgeteilt werden. Dabei entstehen nur einzelne Zeichen. Das Ergebnis für dieses Teil-Raster ist damit (1 0 1 1).

Das (Gesamt-)Ergebnis ist also:

(1 1 1 (1 (1 0 1 1) 1 1))

Das ist Informatik!

Wenn immer mehr Informatiksysteme wie PCs, Smartphones und Tablets benutzt werden, entstehen immer mehr digitale Daten. Und auch wenn die Systeme immer mehr Kapazität zum Speichern und Übertragen digitaler Daten besitzen: es reicht nie aus. Es bleibt also interessant, sich zu überlegen, wie man Texte, Bilder, Audio- oder Videodaten komprimieren kann, also so zu speichern, dass man möglichst wenig Speicherplatz braucht und keine Information verloren geht.

Die Informatik kennt viele Verfahren, Daten verlustfrei zu komprimieren. Das in dieser Aufgabe beschriebene Verfahren funktioniert am besten, wenn große Bereiche gleiche Daten enthalten. Die Aufteilung in vier gleich große Bereiche (und ggf. deren Aufteilung in vier weitere Bereiche usw.) kann man mit Hilfe sogenannter Quadrees im Computer nachvollziehen. Ein Quadtree ist eine besondere Variante der Datenstruktur Baum, die in der Informatik eine besonders wichtige Rolle spielt.



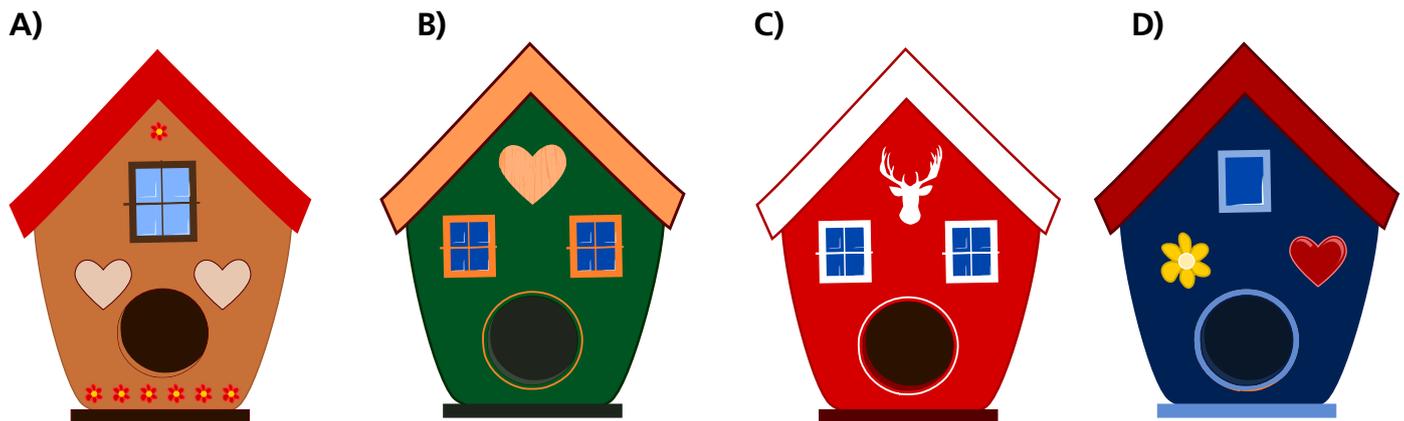
Vogelhaus

Tim wünscht sich zum Geburtstag ein Vogelhaus.

Tim sagt: "Ich möchte ein Vogelhaus mit zwei Fenstern und einem Herzen."

Welches Vogelhaus ist das richtige für Tim?

Der runde Eingang ist kein Fenster.



Antwort B ist richtig:

Dieses Vogelhaus hat zwei Fenster und ein Herz.

Antwort A ist nicht richtig, weil dieses Haus ein Fenster und zwei Herzen hat.

Antwort C ist nicht richtig, weil dieses Haus zwar zwei Fenster, aber kein Herz hat.

Antwort D ist nicht richtig, weil dieses Haus nur ein Fenster hat.

Das ist Informatik!

Wer diese Biberaufgabe richtig beantworten will, muss den Text aufmerksam lesen, verstehen und richtig umsetzen: Tims Wunsch bezieht sich nur auf die Anzahl der Fenster eines Vogelhauses und die Anzahl der darauf abgebildeten Herzen. Deshalb genügt es, nur diese Eigenschaften der Vogelhäuser zu betrachten. Das gelingt am besten, wenn man sich von anderen Eigenschaften nicht ablenken lässt, selbst dann nicht, wenn diese sehr auffällig sind.

Sich auf das Wesentliche zu konzentrieren und Unwichtiges beiseite zu lassen, ist ein wesentlicher Bestandteil informatischen Denkens (engl.: computational thinking): Abstraktion. Wer Informatiksysteme bauen will, kann die unendliche Vielfalt der Wirklichkeit nicht in die Endlichkeit von Computern und Programmen hineinpresse. Man muss sich entscheiden, welche Objekte und welche ihrer Eigenschaften in das System „eingebaut“ werden. Dabei besteht die Gefahr, Wichtiges zu vergessen. Wenn sich als nächstes Tina ein grünes Vogelhaus wünscht, aber nur die Anzahlen der Fenster und Herzen bekannt sind, lässt sich Tinas Wunsch nicht erfüllen.



Wortabstand

Um ein Wort zu ändern, darf man die folgenden Schritte machen:

- einen Buchstaben an einer beliebigen Stelle hinzufügen;
- einen Buchstaben an einer beliebigen Stelle entfernen;
- einen Buchstaben an einer beliebigen Stelle durch einen anderen Buchstaben ersetzen.

Der Abstand zwischen zwei Wörtern ist die Mindestanzahl solcher Schritte, die man benötigt, um das erste Wort in das zweite zu ändern.

Ein Beispiel: Der Abstand zwischen „rennen“ und „stehen“ ist 4:

- | | | | |
|------------|---|---------|--------------------------|
| 1. rennen | → | sennen | („r“ durch „s“ ersetzen) |
| 2. sennen | → | stennen | („t“ einfügen) |
| 3. stennen | → | stehen | („n“ durch „h“ ersetzen) |
| 4. stehen | → | stehen | („n“ entfernen) |

Was ist der Abstand zwischen „Emil“ und „Erich“?

So ist es richtig:

Der Abstand zwischen „Emil“ und „Erich“ ist 3, denn man kann folgende Schritte machen, um „Emil“ in „Erich“ zu ändern:

Emil → Eril → Eric → Erich

Mit weniger als drei Schritten geht es nicht, denn

- „Erich“ ist um einen Buchstaben länger als „Emil“; deshalb wird mindestens ein Hinzufüge-Schritt benötigt.
- In „Erich“ sind weder das „m“ noch das „l“ aus „Emil“ enthalten; deshalb werden zwei Ersetzen-Schritte benötigt.

Das ist Informatik!

Das in dieser Biberaufgabe beschriebene Maß für den Abstand zwischen zwei Wörtern wurde 1965 vom russischen Mathematiker Vladimir Levenshtein erstmals beschrieben und wird deshalb „Levenshtein-Distanz“ genannt. Da die Levenshtein-Distanz mit Hilfe von Operationen zum Bearbeiten bzw. Editieren von Wörtern definiert ist, heißt sie auch „Editierdistanz“.

Dieses Maß wird beispielsweise verwendet, um bei Rechtschreibhilfen korrekte Schreibweisen vorzuschlagen: Wenn die Editierdistanz zwischen dem falsch geschriebenen Wort und der vorgeschlagenen richtigen Schreibweise klein ist, hat man wahrscheinlich nur einen Tippfehler gemacht.

Auch wenn man bei „Editieren“ vorrangig an Wörter, also Folgen von Buchstaben denkt, kann man mit der Editierdistanz und ähnlichen Maßen auch andere Arten von Folgen vergleichen, etwa DNA-Stränge (Folgen von Molekülen) oder Bilder (Folgen von Bildpunkten).

<https://de.wikipedia.org/wiki/Levenshtein-Distanz>



Zerteile den Code

In einem speziellen Code für Texte wird jeder Buchstabe durch ein Codewort aus den Ziffern 0 bis 9 kodiert. Die Besonderheit: Kein Codewort darf mit dem Codewort eines anderen Buchstabens beginnen. Ein Beispiel: Der Buchstabe **X** wird durch **12** kodiert.

Nun kann **Y** durch **2** kodiert werden. Denn **12** beginnt nicht mit **2** (und **2** nicht mit **12**).

Jetzt kann **Z** durch **11** kodiert werden; denn weder **12** noch **2** beginnen mit **11**, und **11** beginnt weder mit **12** noch mit **2**. **21** wäre jedoch nicht als Codewort für **Z** erlaubt, weil es mit **2**, also dem Codewort von **Y** beginnt.

Das Wort **BEBRAS** wird durch die Ziffernfolge unten kodiert.

Teile die Ziffernfolge in die Codewörter der einzelnen Buchstaben!

1 2 1 1 2 2 3 3 3 2 1

So ist es richtig:

1 21 1 22 33 321

Wir betrachten die Ziffernfolge von links. Falls **B** durch **12** kodiert würde, wäre die folgende **1** das Codewort für **E**. Direkt dahinter käme nämlich wieder **12** für das zweite **B**. Doch das ist gegen die Regel: Das Codewort für **B** würde mit dem für **E** beginnen. Längere Anfangsstücke der Ziffernfolge (z.B. **121**, **1211** etc.) können auch nicht Codewörter für **B** sein, weil sie alle nur einmal in der Ziffernfolge enthalten sind. Folglich ist das Codewort für **B** die Ziffer **1**.

Nun muss das Codewort für **E** folgen. Die **2** alleine ist es nicht, denn dann müsste der Anfang der Ziffernfolge zu **BEBB** dekodiert werden. Das Codewort **21** hingegen ist richtig: Erstens wird der Anfang so zu **BEB**, und zweitens ist ein längeres Codewort nicht möglich, da sonst nur noch die letzte **1** das zweite **B** sein kann – aber **BEBRAS** hört nicht mit **B** auf. Nun wissen wir, dass **1 21 1** die Kodierung für **BEB** ist.

Wir machen mit dem Rest weiter: **2233321** ist die Kodierung für **RAS**. Die **2** alleine kann also nicht das Codewort für **R** sein, sonst hätten wir **RR** zu Beginn. Das Codewort für **R** ist also mindestens die **22** beinhalten. Am Ende wiederum sind **1** und **21** schon als Codewörter vergeben. Das Codewort für **S** muss also mindestens die Folge **321** sein. Zwischen **22** und **321** steht **33**. Das muss das Codewort für **A** sein: Das einzig andere noch möglich Codewort wäre **3**. Dann müsste **3321** das Codewort für **S** sein – und würde mit dem Codewort für **A** beginnen; das ist gegen die Regel. Die Aufteilung des hinteren Teils ist also **22 33 321**.

Das ist Informatik!

Der Code, der in dieser Biberaufgabe beschrieben wird, ist ein Beispiel für einen Präfixcode. Ein Präfix ist eine Zeichenfolge zu Beginn einer anderen Zeichenfolge. Bei einem Präfixcode darf kein Codewort Präfix eines anderen Codeworts sein. Das heißt: kein Codewort darf mit einem anderen Codewort beginnen. Bei Präfixcodes haben die Codewörter unterschiedliche Länge. Der Vorteil der Präfix-Regel ist, dass man keine Trennsymbole zwischen Codewörtern benötigt. Man kann immer erkennen, an welcher Stelle das nächste Codewort beginnt. Wenn man kurze Codewörter für häufig vorkommende Buchstaben wählt, kann man Texte sehr effizient kodieren und große Textmengen platzsparend speichern.

Die Huffman-Kodierung ist eine Methode, einen optimalen Präfixcode zu finden. Sie ist weit verbreitet und steckt z.B. hinter bekannten Datenformaten wie JPEG und MP3.

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

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



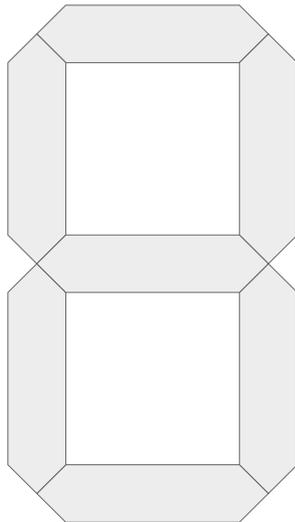
Ziffernsegmente

In einem Haus mit 10 Stockwerken zeigt der Aufzug das aktuelle Stockwerk mit einer 7-Segment-Anzeige an. Die Ziffern 0 (für Erdgeschoss) bis 9 sehen dabei so aus:



Um eine Ziffer anzuzeigen, leuchten also mindestens 2 und höchstens 7 Segmente. Vor Kurzem war ein Segment kaputt und blieb dunkel. Trotzdem konnte man alle Ziffern erkennen und voneinander unterscheiden. Die meisten Segmente müssen aber immer funktionieren, damit man die Ziffern voneinander unterscheiden kann.

Markiere alle Segmente, die immer funktionieren müssen!

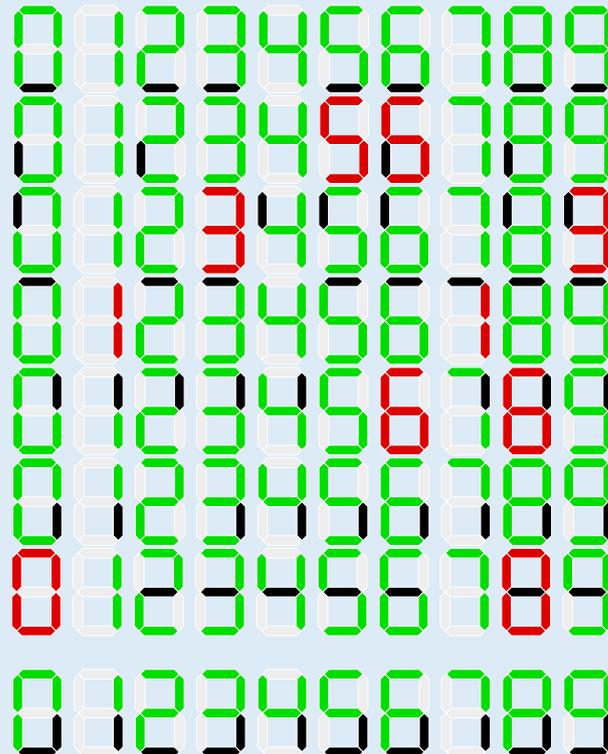




So ist es richtig:



Wenn nach dem Ausfall eines Segmentes mindestens zwei Ziffern nicht mehr unterscheidbar sind, muss dieses Segment immer funktionieren. Im Bild ist in jeder der oberen sieben Zeilen ein bestimmtes Segment abgeschaltet. Die abgeschalteten Segmente sind schwarz eingefärbt. In fünf Zeilen sind dann zwei Ziffern nicht mehr unterscheidbar; diese Ziffern sind rot eingefärbt. Die in diesen Zeilen abgeschalteten Segmente müssen also immer funktionieren.



Heißt das, dass die verbleibenden zwei Segmente nicht immer funktionieren müssen? Das prüfen wir in der letzten Zeile erfolgreich nach. In der letzten Zeile im Bild sind diese beiden Segmente abgeschaltet. Trotzdem sind alle Ziffern unterscheidbar, weil keine zwei gleichen Muster von leuchtenden Segmenten entstehen.

Das ist Informatik!

Es gibt $2^7 = 128$ Möglichkeiten, wie die einzelnen Segmente einer 7-Segment-Anzeige ausfallen können. Nun könnte man versuchen wollen, alle Möglichkeiten durchzuprobieren. Diese Vorgangsweise nennt man Brute-Force-Methode (von englisch „brute force“ für „rohe Gewalt“) oder erschöpfende Suche – durch die Menge möglicher Datenkonstellationen, die potenziell für die Bestimmung einer Lösung interessant sind. Bei den allermeisten realen Informatikproblemen ist aber nicht genügend Zeit, um alle diese Möglichkeiten komplett zu durchsuchen. Man muss also eine Lösungsstrategie entwickeln.

In der obigen Erklärung werden einzelne Segmente isoliert betrachtet. Das Problem wird so in kleinere Teilprobleme zerlegt. Das ist eine der Methoden des informatischen Denkens (englisch: Computational Thinking). In diesem Fall konzentriert man sich zunächst auf einzelne Segmente. Letztendlich muss man aber noch nachprüfen, ob die Kombination der einzelnen gefundenen Segmente tatsächlich eine Lösung darstellt.



Träger:



GESELLSCHAFT
FÜR INFORMATIK



Fraunhofer
IUK-TECHNOLOGIE



max planck institut
informatik

GEFÖRDERT VOM



Bundesministerium
für Bildung
und Forschung