



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

Aufgaben und Lösungen 2019 Schuljahre 3/4

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

Herausgeber:

Christian Datzko, Susanne Datzko, Juraj Hromkovič, Regula Lacher

010100110101011001001001
010000010010110101010011
010100110100100101000101
001011010101001101010011
01001001010010010010001

SV!A

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





Mitarbeit Informatik-Biber 2019

Christian Datzko, Susanne Datzko, Olivier Ens, Hanspeter Erni, Nora A. Escherle, Martin Guggisberg, Saskia Howald, Lucio Negrini, Gabriel Parriaux, Elsa Pellet, Jean-Philippe Pellet, Beat Trachsler.

Herzlichen Dank an:

Juraj Hromkovič, Michelle Barnett, Michael Barot, Anna Laura John, Dennis Komm, Regula Lacher, Jacqueline Staub, Nicole Trachsler: ETHZ

Gabriel Thullen: Collège des Colombières

Valentina Dagienė: Bebras.org

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

Chris Roffey: University of Oxford, Vereinigtes Königreich

Carlo Bellettini, Violetta Lonati, Mattia Monga, Anna Morpurgo: ALaDDIn, Università degli Studi di Milano, Italien

Gerald Futschek, Wilfried Baumann, Florentina Voboril: Oesterreichische Computer Gesellschaft, Österreich

Zsuzsa Pluhár: ELTE Informatikai Kar, Ungarn

Eljakim Schrijvers, Justina Dauksaite, Arne Heijenga, Dave Oostendorp, Andrea Schrijvers, Kyra Willekes, Saskia Zweerts: Cuttle.org, Niederlande

Christoph Frei: Chragokyberneticks (Logo Informatik-Biber Schweiz)

Andrea Leu, Maggie Winter, Brigitte Manz-Brunner: Senarclens Leu + Partner

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

Die französischsprachige Übersetzung wurde von Elsa Pellet und die italienischsprachige Übersetzung von Veronica Ostini erstellt.



INFORMATIK-BIBER SCHWEIZ
CASTOR INFORMATIQUE SUISSE
CASTORO INFORMATICO SVIZZERA

Der Informatik-Biber 2019 wurde vom Schweizerischen Verein für Informatik in der Ausbildung SVIA durchgeführt und von der Hasler Stiftung unterstützt.

HASLERSTIFTUNG

Hinweis: Alle Links wurden am 1. November 2019 geprüft. Dieses Aufgabenheft wurde am 2. Januar 2020 mit dem Textsatzsystem \LaTeX erstellt.



Die Aufgaben sind lizenziert unter einer Creative Commons Namensnennung – Nicht-kommerziell – Weitergabe unter gleichen Bedingungen 4.0 International Lizenz. Die Autoren sind auf S. 28 genannt.



Vorwort

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

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

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

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

Der Informatik-Biber 2019 wurde in fünf Altersgruppen durchgeführt:

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

Die Stufen 3 und 4 hatten 9 Aufgaben zu lösen, jeweils drei davon aus den drei Schwierigkeitsstufen leicht, mittel und schwer. Die Stufen 5 und 6 hatten 12 Aufgaben zu lösen, jeweils vier davon aus den drei Schwierigkeitsstufen leicht, mittel und schwer. Jede der anderen Altersgruppen hatte 15 Aufgaben zu lösen, jeweils fünf davon aus den drei Schwierigkeitsstufen leicht, mittel und schwer.

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

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

Das international angewandte System zur Punkteverteilung soll dem erfolgreichen Erraten der richtigen Lösung durch die Teilnehmenden entgegenwirken.

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

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

Bei vielen Aufgaben wurden die Antwortalternativen am Bildschirm in zufälliger Reihenfolge angezeigt. Manche Aufgaben wurden in mehreren Altersgruppen gestellt.



Für weitere Informationen:


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

Informatik-Biber

Nora A. Escherle

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

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

 <https://www.facebook.com/informatikbiberch>



Inhaltsverzeichnis

Mitarbeit Informatik-Biber 2019	i
Vorwort	ii
Inhaltsverzeichnis	iv
1. Sommer, Sonne, Badi!	1
2. Kratzbilder	5
3. Kiosk	9
4. Beavercoins	11
5. Rauchsignale	13
6. Stempel	15
7. Welcher Turm?	19
8. Auf Reisen durchs Weltall	21
9. Zeichenroboter	25
A. Aufgabenautoren	28
B. Sponsoring: Wettbewerb 2019	29
C. Weiterführende Angebote	32



1. Sommer, Sonne, Badi!

Es ist Sommer und die zwölfjährige Anita möchte in der Badi schwimmen gehen. Sie nimmt ihren sechsjährigen Bruder Hans mit.

Beim Eingang der Badi steht folgende Regel:

- Mindestalter 8 Jahre; Kinder unter 8 Jahren nur in Begleitung einer Person, die älter als 10 Jahre ist.

Wer darf in die Badi?

- A) Anita und Hans.
- B) Anita, aber Hans nicht.
- C) Anita nicht, aber Hans.
- D) Weder Anita noch Hans.





Lösung

Die Regel hat zwei Bedeutungen:

1. Alle Personen, die 8 Jahre oder älter sind, dürfen in die Badi. Da Anita älter als 8 Jahre ist, darf sie in die Badi.
2. Personen, die jünger als 8 Jahre sind, dürfen dann in die Badi, wenn sie von einer Person begleitet werden, die älter als 10 Jahre ist. Da Hans von Anita begleitet wird und Anita älter als 10 Jahre ist, darf auch Hans in die Badi.

Die richtige Antwort also A) Anita und Hans.

Dies ist Informatik!

Die Regel der Badi stellt *Bedingungen* auf, unter denen etwas erlaubt oder verboten ist. In diesem Fall werden zwei Bedingungen aufgestellt, unter denen eine Person hinein darf oder nicht. *Wenn* die Bedingung erfüllt ist, *dann* darf die Person in die Badi. Man hätte die Regel also auch so formulieren können:

Wenn die Person 8 Jahre oder älter ist:

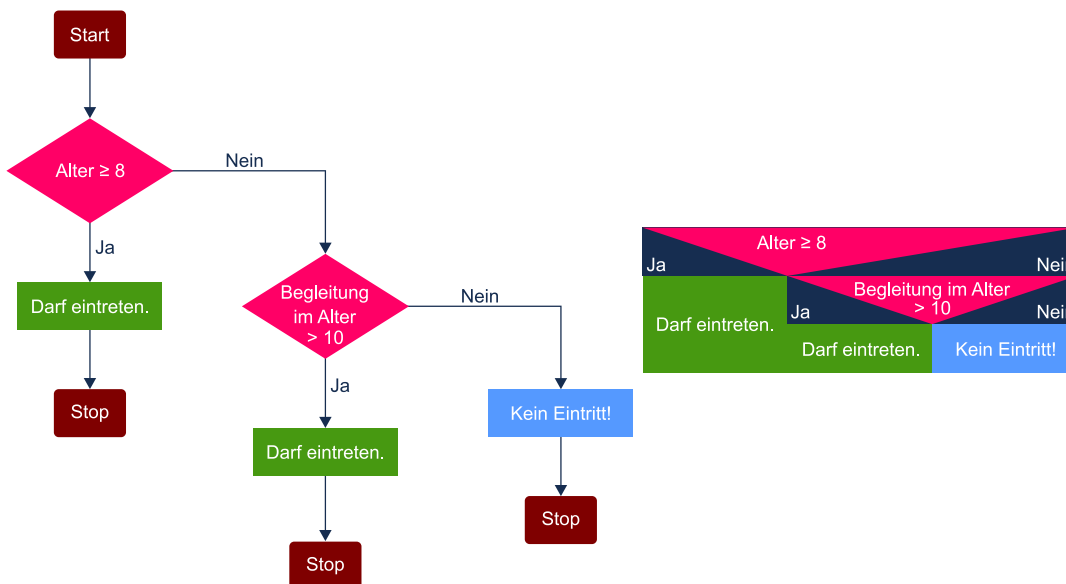
Dann darf sie in die Badi

Sonst: Wenn die Person von einer älter als 10 Jahre alten Person begleitet wird:

Dann darf sie in die Badi

Sonst darf sie nicht in die Badi

Man kann das auch als *Flussdiagramm* und als *Struktogramm* darstellen:



Solche Entscheidungsvorgänge nennt man in der Informatik *Verzweigungen*. Sie kommen dort sehr häufig vor.



Stichwörter und Webseiten

Verzweigung, Flussdiagramm, Struktogramm

- https://de.wikipedia.org/wiki/Bedingte_Anweisung_und_Verzweigung
- <https://de.wikipedia.org/wiki/Programmablaufplan>
- <https://de.wikipedia.org/wiki/Nassi-Shneiderman-Diagramm>





2. Kratzbilder

Mit Kratzpapier kannst du einfach bunte Bilder malen. Mit Hilfe eines Holzstabes entfernst Du die oberste Schicht und der farbige Untergrund wird sichtbar.

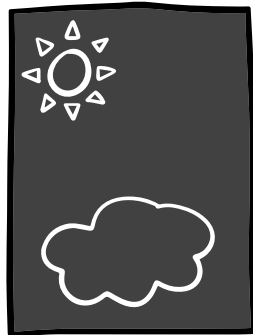
<p>Am Anfang ist das Kratzpapier überall schwarz beschichtet.</p>	<p>Hinter der schwarzen Schicht sind diese vier Farben versteckt.</p>	<p>Mit dem Holzstab wurde ein Teil der schwarzen Schicht weggekratzt. Dort siehst du die dahinter versteckten Farben.</p>

Bei welchem der vier Bilder erscheinen beim Zeichnen genau drei Farben?

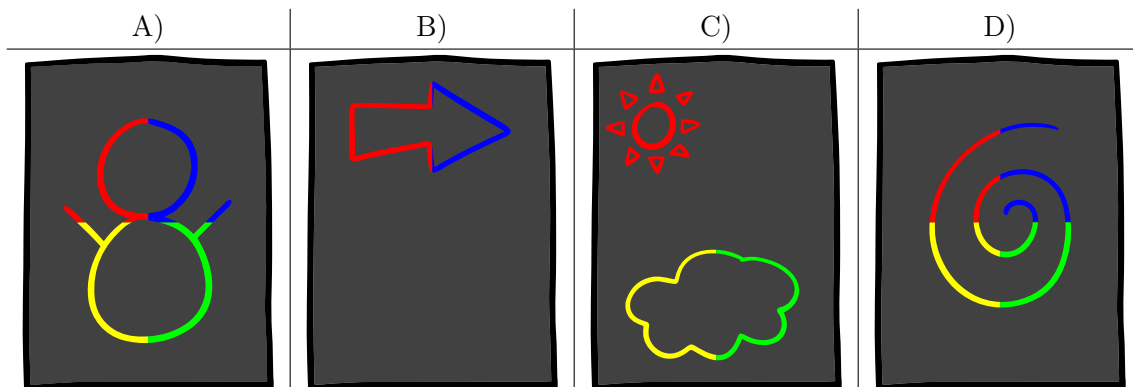
A)	B)	C)	D)



Lösung



Die richtige Antwort ist C)
Beim Zeichnen der vier Bilder kommen die folgenden Farben hervor:



Die korrekte Antwort ist daher C): die Farben Rot, Gelb und Grün erscheinen. Die vierte Farbe blau erscheint nicht, da das obere rechte Viertel frei bleibt.
Bei den Antworten A) und D) erscheinen alle vier Farben, bei der Antwort B) nur die beiden Farben rot und blau.

Dies ist Informatik!

Durch das Entfernen der obersten Schicht von Kratzpapier wird diese an der Stelle *transparent*, man kann also durch die Ebene (manchmal auch *layer* genannt) durchsehen und sieht die Farbe dahinter. In vielen Bildbearbeitungsprogrammen werden Ebenen, die an einigen Stellen transparent sind, häufig verwendet. Meistens nutzt man sie aber umgekehrt: man hat beispielsweise ein Foto im Hintergrund und fügt darüber als neue Ebene einen Text hinzu. Diese Ebene ist dann überall transparent ausser dort, wo der Text ist. Natürlich könnte man auch mit dem Text das Bild direkt überschreiben. Wenn man aber mehrere Ebenen verwendet, kann man nachher eine Ebene ändern, und alle anderen Ebenen bleiben gleich.

In dieser Aufgabe musste man sich vorstellen, wie die hintere Schicht aussieht, wenn man die vordere Schicht transparent macht. Das ist viel einfacher, wenn man sich das Bild in kleinere Bilder zerlegt. Dann kann man sich für jedes Viertel überlegen, ob die vordere Schicht irgendwo transparent ist oder nicht. Man weiss dann, ob die entsprechende Farbe im Gesamtbild vorkommt oder nicht. Diesen Vorgang nennt man *Dekomposition*, er kommt in der Informatik häufig vor.

Stichwörter und Webseiten

Bildbearbeitung, Ebenentechnik, Dekomposition



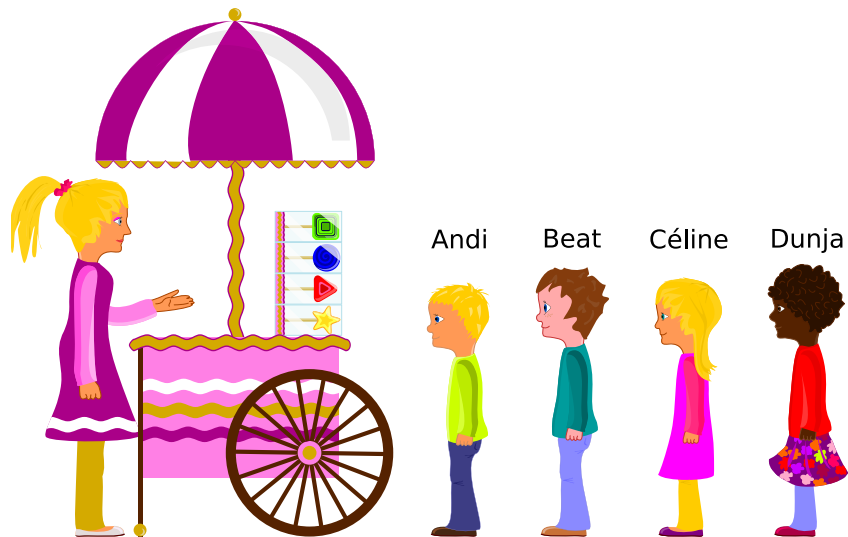
- <https://de.wikipedia.org/wiki/Ebenentechnik>
- [https://en.wikipedia.org/wiki/Decomposition_\(computer_science\)](https://en.wikipedia.org/wiki/Decomposition_(computer_science))





3. Kiosk

Andi, Beat, Céline und Dunja stehen in einer Warteschlange beim Kiosk an. Die Verkäuferin hat einen Stapel Schleckstengel vor sich. Sie verkauft immer den oberste Schleckstengel des Stapels. Andi bekommt den grünen, viereckigen Schleckstengel 🍭, denn er ist der erste in der Warteschlange und bekommt somit den obersten Schleckstengel.



Wer bekommt den roten, dreieckigen Schleckstengel 🍭?

- A) Andi
- B) Beat
- C) Céline
- D) Dunja



Lösung

Andi ist zuerst dran: er erhält wie bereits in der Aufgabe beschrieben den obersten grünen, viereckigen Schleckstengel .

Nachdem der grüne, viereckige Schleckstengel verkauft ist, ist neu der blaue, runde Schleckstengel zuoberst. Diesen erhält Beat als zweiter in der Warteschlange.

Nach dem blauen, runden Schleckstengel ist neu der rote, dreieckige Schleckstengel zuoberst. Diesen erhält Céline als dritte in der Warteschlange. Damit ist die richtige Antwort C) Céline.

Dunja geht ebenfalls nicht leer aus: für sie bleibt der gelbe, sternförmige Schleckstengel .

Dies ist Informatik!

Andi, Beat, Céline und Dunja warten in einer *Warteschlange*. Wenn Eddie sich anstellen möchte, würde er sich *hinten* nach Dunja anstellen müssen. Als erster jedoch kommt Andi von *vorne* an der Warteschlange an die Reihe.

Die Schleckstengel jedoch sind auf einem *Stapel* gelegt. Wenn die Verkäuferin einen weiteren Schleckstengel verkaufen möchte, würde sie ihn *oben auf* den Stapel legen. Sie verkauft auch immer den Schleckstengel, den sie *von oben* vom Stapel nimmt.

Eine Warteschlange (engl. *queue*) kann Elemente hinten hinzufügen (engl. *enqueue*) und vorne entfernen (engl. *dequeue*). Sie funktioniert also nach dem Prinzip, dass das zuerst hinzugefügte Element zuerst entfernt wird (engl. *“First-In-First-Out”*, kurz *“FIFO”*). Ein *Stapel* (engl. *stack*) kann Elemente oben hinzufügen (engl. *push*) und oben entfernen (engl. *pop*). Er funktioniert also nach dem Prinzip, dass das zuletzt hinzugefügte Element zuerst entfernt wird (engl. *“Last-In-First-Out”*, kurz *“LIFO”*). Häufig bieten Warteschlange und Stapel noch zusätzlich Funktionen, um das nächste zu entfernende Element anzuzeigen (engl. *front* respektive *top*), oder um abzufragen, ob die Warteschlange oder der Stapel leer ist (engl. *empty*).

Diese beiden Strukturen werden häufig in Computern verwendet, weil sie so einfach sind. Stapel beispielsweise wurden bereits 1945 von Konrad Zuse in einem der ersten Computer der Welt verwendet.

Stichwörter und Webseiten

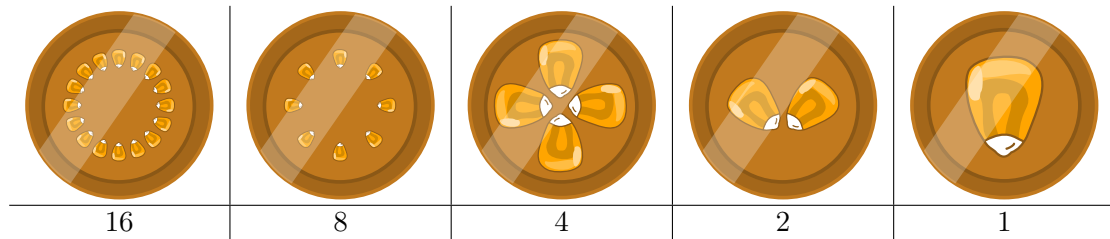
Warteschlange, Stapel

- [https://de.wikipedia.org/wiki/Warteschlange_\(Datenstruktur\)](https://de.wikipedia.org/wiki/Warteschlange_(Datenstruktur))
- <https://de.wikipedia.org/wiki/Stapelspeicher>
- https://de.wikipedia.org/wiki/Zuse_Z4



4. Beavercoins

Im Biberland verwendet man „Beavercoins“ als Wahrung. Die Munzen haben die folgenden Werte:






Die Biber tragen nicht gerne viele Munzen bei sich und zahlen deswegen mit so wenig Munzen wie moglich.

Mit welchen Munzen wurdest Du 13 Beavercoins bezahlen, wenn Du moglichst wenige Munzen verwendest?



Lösung

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

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

Dies ist Informatik!

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

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

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

Stichwörter und Webseiten





Binäres Zahlensystem, Abakus

- <https://de.wikipedia.org/wiki/Dualsystem>
- [https://de.wikipedia.org/wiki/Abakus_\(Rechenhilfsmittel\)](https://de.wikipedia.org/wiki/Abakus_(Rechenhilfsmittel))



5. Rauchsignale

Ein Biber sitzt immer oben auf dem Berg und beobachtet das Wetter. Er übermittelt den Bibern im Tal, wie das Wetter werden wird. Er nutzt dazu Rauchsignale, die aus fünf nacheinander folgenden Rauchwolken bestehen. Eine Rauchwolke ist entweder klein oder gross. Die Biber haben folgende Rauchsignale vereinbart:

			
Es wird gewittrig.	Es wird regnerisch.	Es wird bewölkt.	Es wird sonnig.

An einem windigen Tag können die Biber im Tal die Rauchwolken nicht gut erkennen. Sicher sind sie sich nur, dass die zweite und vierte Rauchwolke gross ist, die anderen haben sie durch ein Fragezeichen ersetzt:



Was könnte das bedeutet haben?

- A) „Es wird gewittrig“ oder „Es wird regnerisch“.
- B) „Es wird regnerisch“ oder „Es wird bewölkt“.
- C) „Es wird regnerisch“ oder „Es wird sonnig“.
- D) „Es wird gewittrig“ oder „Es wird bewölkt“.



Lösung

Die Biber im Tal haben an der zweiten und an der vierten Stelle eine grosse Rauchwolke erkannt. Bei den Rauchsignal „Es wird gewittrig“ und „Es wird bewölkt“ sind an diesen beiden Stellen grosse Rauchwolken, also an der zweiten und der vierten Stelle. Bei „Es wird regnerisch“ und „Es wird sonnig“ sind an diesen Stellen kleine Rauchwolken, damit passen diese Rauchsignal nicht zu der Beobachtung der Biber im Tal.

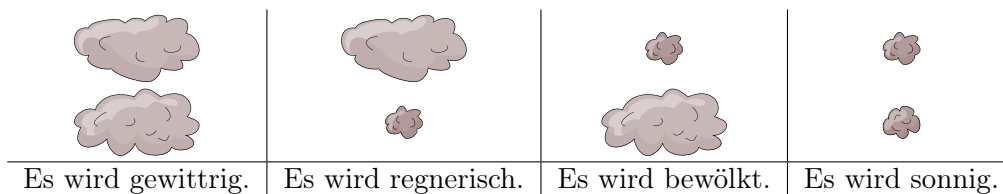
Deshalb ist die richtige Antwort D) „Es wird gewittrig“ oder „Es wird bewölkt“.

Dies ist Informatik!

Wenn man eine Nachricht übermitteln muss, möchte man, dass die Nachricht richtig beim Empfänger ankommt. Die Nachricht in dieser Aufgabe wird mit Hilfe von grossen und kleinen Rauchwolken übermittelt. Im allgemeinen Fall spricht man von *Symbolen*. Daher ist es sinnvoll, eine Folge von Symbolen so zu wählen, dass die zu übermittelnde Nachricht auch dann verstanden werden kann, wenn sie unterwegs beschädigt wurde. Dies kann man erreichen, indem man mehr Information kommuniziert als absolut notwendig. Man nennt diese zusätzliche Information *redundant*.

Wenn man die beschädigte Nachricht mit höchstens n Fehlern rekonstruieren kann, spricht man von n -selbstkorrigierenden Kodierungen. Nachrichten als Folgen von Symbolen so darzustellen, dass man die Nachrichten rekonstruieren kann, auch wenn ihre Darstellung unterwegs beschädigt wurde, ist eine typische Aufgabe für Informatiker. Sie ermöglichen so zum Beispiel, Musik von CDs oder Videos von DVDs korrekt abzuspielen, auch wenn bei der Übertragung einige Fehler aufgetreten sind.

Für diese Aufgabe hätten übrigens zwei Rauchwolken genügt, um die vier unterschiedlichen Nachrichten zu übermitteln:



Die Biber verwenden aber fünf Rauchwolken. Das erlaubt ihnen in Fällen wo zwei oder in einigen Fällen sogar drei Rauchwolken „unlesbar“ sind, die Nachricht trotzdem richtig zu verstehen. Die Biber haben sich die Nachrichten übrigens so überlegt, dass sich je zwei Nachrichten an mindestens drei Stellen unterscheiden.

Stichwörter und Webseiten

Fehlerkorrekturverfahren

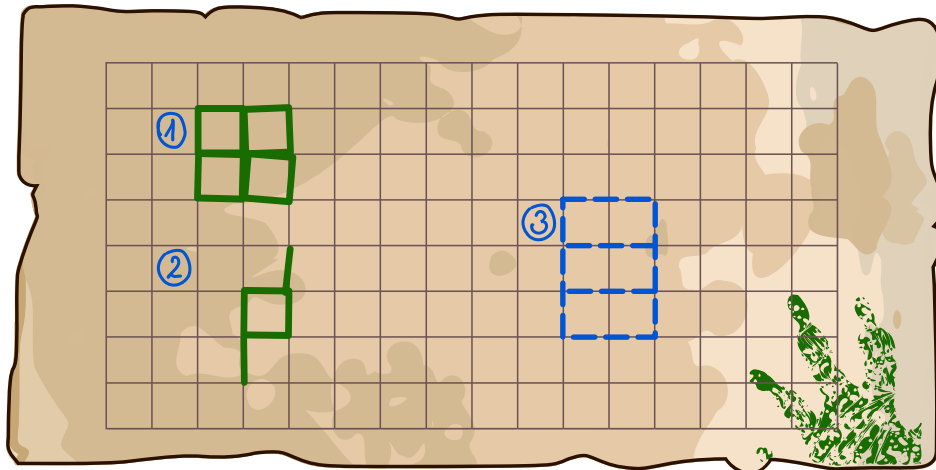
- <https://de.wikipedia.org/wiki/Fehlerkorrekturverfahren>



6. Stempel

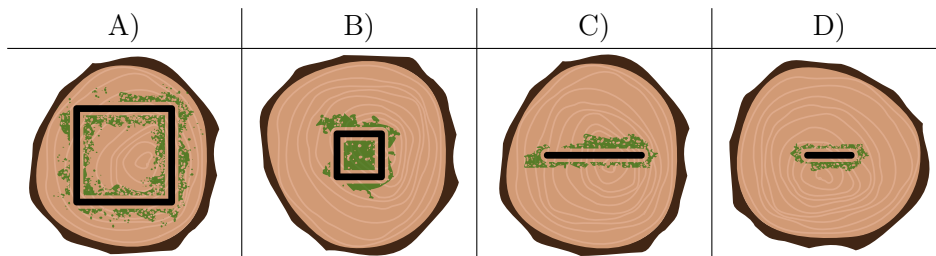
Der Biber Paul hat vier Stempel A, B, C und D, wie unten gezeigt. Mit diesen Stempeln hat Paul die beiden Figuren ① und ② gestempelt.

- Für die Figur ① hat Paul den Stempel B viermal verwendet.
- Für die Figur ② hat Paul einmal den Stempel B und zweimal den Stempel D verwendet.



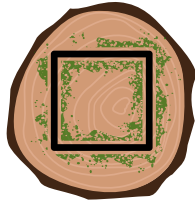
Nun möchte Paul die Figur ③ haben. Pauls Schwester Maria behauptet, dass sie für die Figur nur zweimal stempeln muss.

Welchen Stempel würde Maria verwenden?

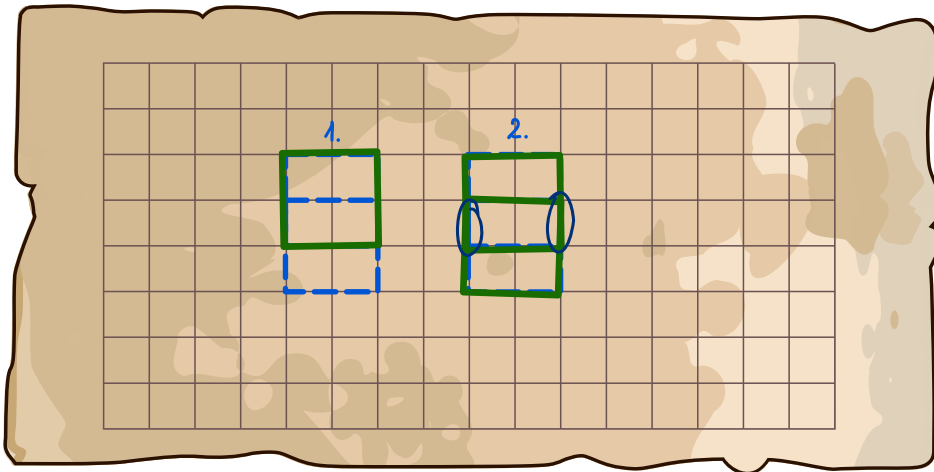




Lösung



Die richtige Antwort ist A) . Wenn Maria ein grosses Quadrat gestempelt hat und den Stempel um ein Kästchen nach oben oder unten verschiebt, entsteht genau die gewünschte Figur. Dabei überschneiden sich zwar zwei Linienteile, aber wenn sie sauber stempelt, sieht man das nicht:



Wenn man die anderen Stempel nur zweimal verwendet, kann man die gewünschte Figur nicht erreichen:

- Mit dem Stempel B ist es unmöglich, ein Rechteck zu stempeln, das zwei Kästchen breit ist und keine Mittellinie hat.
- Mit dem Stempel C könnte sie die Figur zwar stempeln, da die Figur aber Linien in der Gesamtlänge von vierzehn Kästchenlängen hat und sie pro Stempelvorgang nur zwei Kästchenlängen stempeln kann, bräuchte sie mindestens sieben Stempelvorgänge. Wenn man genau schaut, stellt man fest, dass sie sogar acht Stempelvorgänge braucht, denn um die beiden senkrechten Striche zu stempeln, braucht sie jeweils zwei Stempelvorgänge (mit Überlappungen) zusätzlich zu den vier waagerechten Linien.
- Mit dem Stempel D könnte sie die Figur zwar stempeln, da die Figur aber Linien in der Gesamtlänge von vierzehn Kästchenlängen hat und sie pro Stempelvorgang nur eine Kästchenlänge stempeln kann, bräuchte sie vierzehn Stempelvorgänge.

Dies ist Informatik!

Für viele Aufgaben gibt es viele verschiedene Lösungen, die alle zum Ziel führen. Oftmals sind einige davon schnell zu finden wie beispielsweise eine Lösung mit Hilfe der Stempel C oder D. Aber nicht alle Lösungen sind gleich „gut“: die verschiedenen Lösungen unterscheiden sich beispielsweise deutlich in der Anzahl der Stempelvorgänge.

Eine der Aufgaben der Informatik ist es, unter vielen möglichen Lösungen einer Aufgabe diejenigen zu finden, die am „besten“ ist. Das ist praktisch sehr wichtig: Wenn man dieselbe Aufgabe in einer



Stunde anstelle von in einem Tag lösen kann, hat man viele Stunden übrig, in denen man sich mit anderen Aufgaben beschäftigen kann.

Zum Messen der Effizienz analysieren Informatiker Vorgänge und beschreiben ihre Dauer in Abhängigkeit von der Menge oder Grösse der zu verarbeitenden Daten. Wenn man beispielsweise in einem sortierten Feld mit 1'000'000 Einträgen nach einem Eintrag sucht, kann man entweder Feld für Feld anschauen und im Schnitt 500'000 Vergleiche machen oder in der Mitte anfangen und immer in der jeweiligen Hälfte weitersuchen ... und hat dann nach spätestens 20 Vergleichen den Eintrag gefunden! Bei drei Sekunden pro Vergleich wäre das ein Unterschied zwischen 17 Tagen ununterbrochener Suche oder 1 Minute.

Stichwörter und Webseiten

Effizienz, Komplexitätstheorie

- [https://de.wikipedia.org/wiki/Effizienz_\(Informatik\)](https://de.wikipedia.org/wiki/Effizienz_(Informatik))
- <https://de.wikipedia.org/wiki/Komplexit%C3%A4tstheorie>

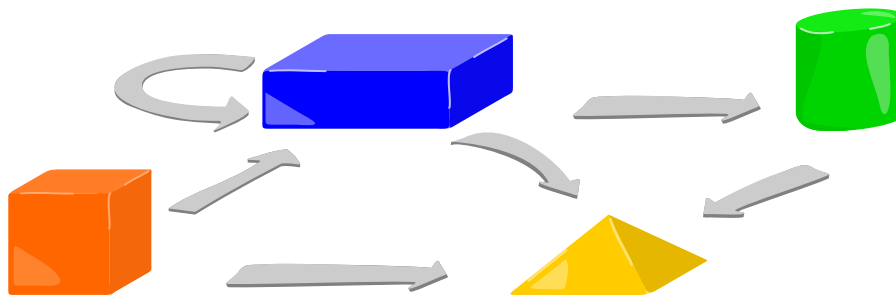




7. Welcher Turm?

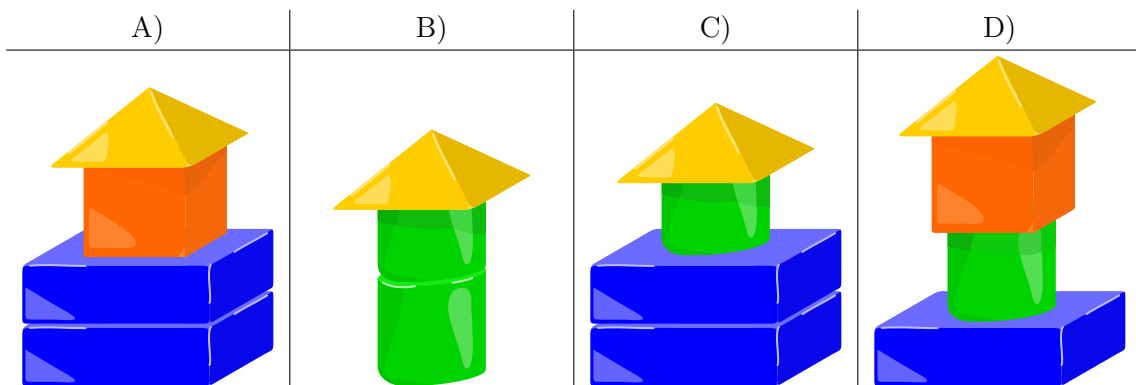
Leons kleine Schwester hat Regeln aufgestellt, wie Bauklötze gestapelt werden können. Sie hat diese in einer Zeichnung mit Pfeilen festgehalten. Ausserdem gilt:

- Leon darf mit einem beliebigen Bauklotz starten.
- Leon muss immer den Pfeilen folgen. Wenn von einem Bauklotz mehrere Pfeile wegführen, darf er einen auswählen. Wenn ein Pfeil zurück zu demselben Bauklotz führt, darf er einen weiteren von dieser Sorte auf den Turm stapeln.
- Leon muss aufhören, wenn kein Pfeil mehr von dem zuletzt gelegten Bauklotz wegführt.



Leon stapelt vier verschiedene Türme für seine kleine Schwester auf.

Welchen der vier Türme hat er nach den Regeln seiner kleinen Schwester gebaut?





Lösung

Der Turm der Antwort A) beginnt korrekt mit zwei blauen Quadern. Nach dem zweiten blauen Quader folgt jedoch ein oranger Würfel, es gibt aber keinen Pfeil von den blauen Quadern zu den orangen Würfeln. Damit ist die Antwort A) falsch.

Der Turm der Antwort B) beginnt korrekt mit einem grünen Zylinder. Nach dem grünen Zylinder folgt jedoch ein weiterer grüner Zylinder, es gibt aber keinen Pfeil von den grünen Zylindern zurück zu den grünen Zylindern. Damit ist die Antwort B) falsch.

Der Turm der Antwort C) beginnt korrekt mit zwei blauen Quadern. Nach dem zweiten blauen Quader folgt korrekt ein grüner Zylinder und nach dem grünen Zylinder korrekt eine gelbe Pyramide. Da von der gelben Pyramide keine Pfeile wegführen, ist es korrekt, dass keine weiteren Bauklötze darauf gestapelt wurden. Damit ist die Antwort C) richtig.

Der Turm der Antwort D) beginnt korrekt mit einem blauen Quader. Nach dem blauen Quader folgt korrekt ein grüner Zylinder. Nach dem grünen Zylinder folgt jedoch ein oranger Würfel, es gibt aber keinen Pfeil von den grünen Zylindern zu den orangen Würfeln. Damit ist die Antwort D) falsch.

Dies ist Informatik!

Die Regeln für den Turmbau basieren darauf, dass der jeweils oberste Bauklotz des Turmes darüber entscheidet, welche Bauklötze als nächstes erlaubt sind. Damit ist der oberste Bauklotz des Turmes der *aktuelle Zustand* des Turmes. Die Regeln legen fest, in welchen nächsten Zustand der Turm *übergangen* kann. Die Graphik mit den Pfeilen ist somit ein *Zustandsdiagramm* oder auch ein *Zustandsübergangsdigramm*. Da alle Bauklötze als unterster Bauklotz verwendet werden können, sind alle Bauklötze mögliche *Startzustände*. Die gelbe Pyramide ist als einziger Bauklotz ein *Endzustand*, mit dem der Turm beendet wird (wenn er nicht vorher umgefallen ist). Die Entscheidung, einen weiteren Bauklotz auf den Turm zu stapeln, ist eine *Eingabe* des Erbauers.

Diese Aspekte des Turmbaus beschreiben einen sogenannten *nichtdeterministischen endlichen Automaten*. Er heisst deswegen nichtdeterministisch, weil es Zustände gibt, in denen verschiedene nächste Wege gewählt werden können: nach einem blauen Quader kann ein weiterer blauer Quader, ein grüner Zylinder oder eine gelbe Pyramide folgen. Er heisst endlich, weil es nur eine endliche Menge von möglichen Zuständen gibt: einer der vier Bauklötze als oberster Bauklotz des Turmes. Theoretisch könnte man mit ihm zwar einen unendlich hohen Turm bauen . . . aber dafür bräuchte man unendlich viele blaue Quader, und hohe Türme fallen manchmal (oftmals zur grossen Freude der Erbauer) um. Das Modell der nichtdeterministischen endlichen Automaten wird in der Informatik häufig verwendet. Sie eignen sich gut zur Beschreibung ganz unterschiedlicher Dinge: das Verhalten von Softwaremodulen oder auch ganzer Programme, einfache sprachliche Strukturen, das Zusammenspiel von Hardware-Bauteilen und vieles mehr. Mit Hilfe einer solchen formalen Beschreibung kann dann auch getestet werden, ob die Software sich wie gewünscht verhält . . . oder ob der Turm richtig gebaut ist.

Stichwörter und Webseiten

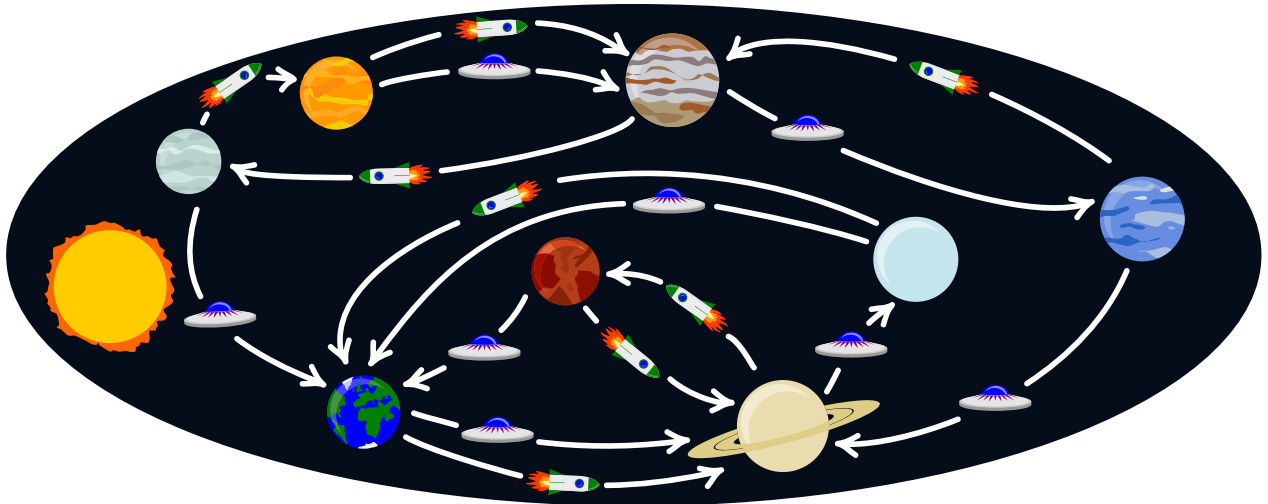
Nichtdeterministische endliche Automaten

- https://de.wikipedia.org/wiki/Nichtdeterministischer_endlicher_Automat
- <https://www.swisseduc.ch/informatik/karatojava/kara/index.html>
- <https://educ.ethz.ch/unterrichtsmaterialien/informatik/kara.html>



8. Auf Reisen durchs Weltall

Astronauten können zwischen den Planeten unseres Sonnensystems mit Raketen 🚀 oder UFOs 🛸 fliegen. Die folgende Karte stellt die möglichen Flugrouten dar:



Ein Astronaut, der von der Venus 🟠 zum Saturn 🪐 reisen will, kann mit einer Rakete 🚀 oder mit einem UFO 🛸 zum Jupiter 🪛 fliegen. Danach kann er mit einem UFO 🛸 zum Neptun 🌌 fliegen und am Schluss mit einem UFO 🛸 zu seinem Zielplaneten Saturn 🪐. Wenn der Astronaut also zuerst mit einer Rakete und anschliessend mit zwei UFOs fliegt, beschreibt er die Reise so:
















Die Astronautin Heidi ist momentan auf dem Planeten Neptun 🌌 und möchte zurück zur Erde 🌍 reisen. Das Raumfahrt-Reisebüro schickt ihr vier Vorschläge.
Welcher der Vorschläge bringt Heidi nicht zurück zur Erde 🌍?

- A) 🛸 🛸 🚀
- B) 🚀 🛸 🚀 🛸
- C) 🚀 🛸 🛸 🛸 🚀
- D) 🚀 🚀 🛸
















Lösung

Die Antwort B)     ist als einzige falsch. Falls Heidi diesen Vorschlag befolgt, landet sie am Schluss wieder auf dem Neptun . Zuerst fliegt sie nämlich mit einer Rakete  zum Jupiter , dann mit einem UFO  zurück zum Neptun , dann wieder mit einer Rakete  zum Jupiter  und zum Schluss wieder mit einem UFO  zurück zum Neptun .



Die anderen drei Vorschläge bringen sie alle zurück zur Erde .

Antwort A): Vom Neptun  mit einem UFO  zum Saturn , mit einem UFO  zum Uranus  und mit einer Rakete  zur Erde .

Antwort C): Vom Neptun  mit einer Rakete  zum Jupiter , mit einem UFO  zum Neptun , mit einem UFO  zum Saturn , mit einem UFO  zum Uranus  und mit einer Rakete  zur Erde .

Antwort D): Vom Neptun  mit einer Rakete  zum Jupiter , mit einer Rakete  zum Merkur  und mit einem UFO  zur Erde .

Dies ist Informatik!

Die Karte der möglichen Flugrouten von Planet zu Planet hat eine besondere Eigenschaft: von jedem Planeten führen genau zwei mögliche Flugrouten weg, eine mit einer Rakete  und eine mit einem UFO . Damit ist immer klar, zu welchem Planeten man als nächstes fliegt, wenn man angibt, mit welchem Transportmittel man fliegt.

Ein solche Karte beschreibt einen *deterministischen endlichen Automaten*. Ein solcher Automat besteht aus einer Menge von möglichen *Zuständen* (in diesem Fall sind das die Namen der Planeten als Standort eines Astronauten), einer Menge von möglichen *Zustandsübergängen* (in diesem Fall sind das die Pfeile auf der Karte mit deren Hilfe der Astronaut seinen Standort wechseln kann) basierend auf bestimmten *Eingaben* („Rakete“ oder „UFO“) sowie einem *Startzustand* (in diesem Fall der Zustand „Neptun“) und einer Menge von *Endzuständen* (in diesem Fall alleine der Zustand „Erde“). Die Karte nennt man dann auch ein *Zustandsdiagramm* oder *Zustandsübergangsdigramm*. Deterministische endliche Automaten werden an vielen Stellen verwendet, weil sie sehr einfach zu programmieren sind. Typische Beispiele sind Kaffeemaschinen, Abwaschmaschinen oder auch Getränkeautomaten. Sie werden aber auch benutzt, um Wörter korrekt zu erkennen (beispielsweise ob ein eingegebener Text eine E-Mail-Adressen darstellt). Man kann endliche Automaten auch mit einer bestimmten Klasse von Grammatiken (sogenannte *reguläre Grammatiken*) und einer bestimmten Klasse künstlicher Sprachen (sogenannten *regulären Sprachen*) in Verbindung bringen und von einer „Welt“ in die andere springen. Das hilft, viele Probleme zu lösen.

Das Raumfahrt-Reisebüro hat übrigens eine andere Aufgabe: es muss in dem Zustandsdiagramm einen möglichen Weg von einem Zustand zu einem anderen finden. Da hilft es, das Zustandsdiagramm als *gerichteten Graphen* anzuschauen und einen möglichen Weg von einem *Knoten* zu einem anderen Knoten mit Hilfe der vorgegebenen *Kanten* suchen. Hierfür gibt es Standard-Algorithmen, so dass das Raumfahrt-Reisebüro nicht immer von vorne anfangen muss zu suchen ...

Stichwörter und Webseiten

Deterministische endliche Automaten, Graphen

- https://de.wikipedia.org/wiki/Deterministischer_endlicher_Automat
- <https://www.swisseduc.ch/informatik/karatojava/kara/index.html>
- <https://educ.ethz.ch/unterrichtsmaterialien/informatik/kara.html>



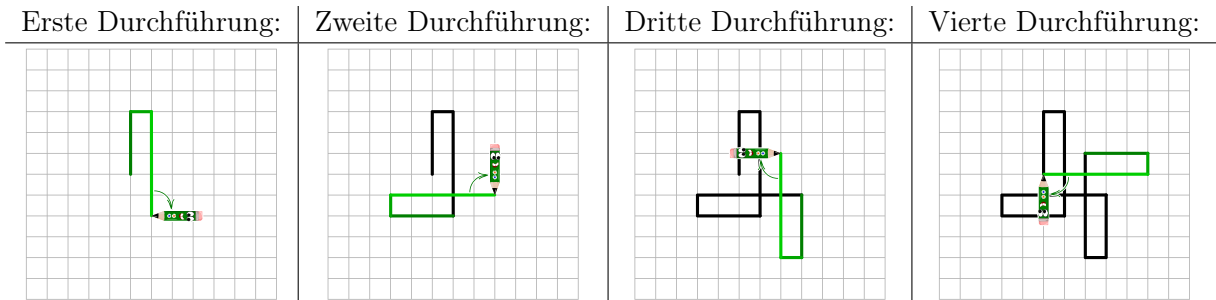
- [https://de.wikipedia.org/wiki/Graph_\(Graphentheorie\)](https://de.wikipedia.org/wiki/Graph_(Graphentheorie))





9. Zeichenroboter

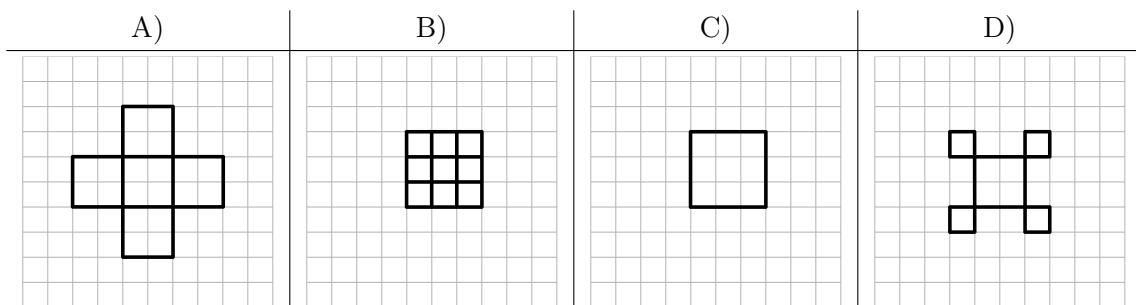
Ein Roboter bewegt sich über ein Raster und zeichnet dabei Linien. Er kann mit Hilfe von drei Zahlen gesteuert werden. Wenn man ihm die Zahlen $3 \nearrow 1 \nearrow 5 \nearrow$ gibt, so zeichnet er die folgende Figur:



Dabei wiederholt er die folgenden Schritte vier Mal:

- Gehe so viele Felder im Raster vor wie die erste Zahl vorgibt.
- Drehe Dich eine Vierteldrehung nach rechts.
- Gehe so viele Felder im Raster vor wie die zweite Zahl vorgibt.
- Drehe Dich eine Vierteldrehung nach rechts.
- Gehe so viele Felder im Raster vor wie die dritte Zahl vorgibt.
- Drehe Dich eine Vierteldrehung nach rechts.

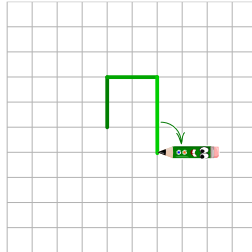
Dem Roboter werden die Zahlen $2 \nearrow 2 \nearrow 3 \nearrow$ gegeben. Wie sehen die gezeichneten Linien aus?



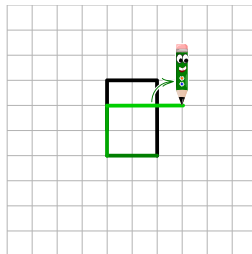


Lösung

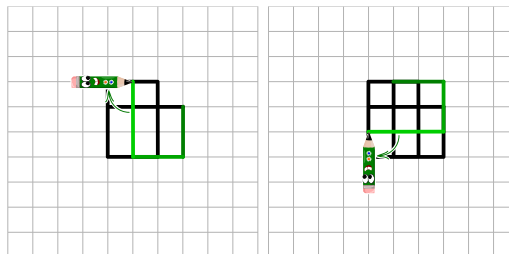
Die Zahlen $2\curvearrowright 2\curvearrowright 3\curvearrowright$ bedeuten, dass der Roboter zunächst 2 Felder vorgeht, sich eine Vierteldrehung nach rechts dreht, wieder 2 Felder vorgeht, sich eine Vierteldrehung nach rechts dreht, dann 3 Felder vorgeht, und sich noch einmal eine Vierteldrehung nach rechts dreht. Danach hat er folgende Linien gezeichnet:



Wenn er dies wiederholt, hat er insgesamt die folgenden Linien gezeichnet:

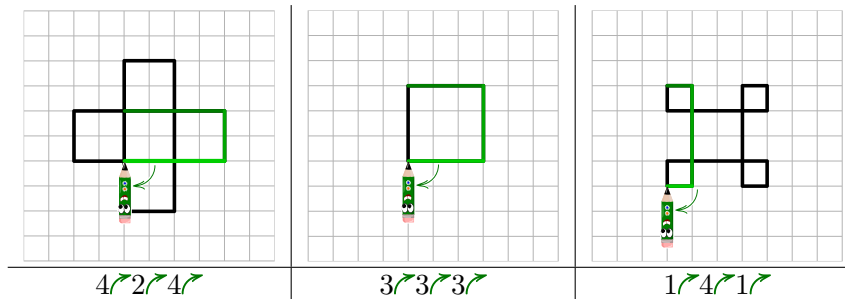


Nach zwei weiteren Wiederholungen sieht das Bild so aus:



Damit ist die Antwort B) richtig.

Die anderen drei Figuren lassen sich übrigens auch mit Hilfe des Roboters zeichnen, sie benötigen einfach andere Zahlen:



Dies ist Informatik!

Der Zeichenroboter in dieser Aufgabe kann nur sehr einfache Programme ausführen. Die Programmiersprache, die der Roboter versteht, kennt nur drei Zahlen als Anweisungen. Jedes Programm



muss genau aus solchen drei Zahlen bestehen, gefolgt von dem Drehsymbol ↻. Ausserdem ist fest eingebaut, dass der Roboter die in einem Programm enthaltenen Anweisungen vier Mal wiederholt, ob dies nun gewollt ist oder nicht.

Die meisten Roboter und Computer verstehen sehr viel komplexere (Programmier-)Sprachen. Die meisten dieser Sprachen haben die gleichen grundlegenden Eigenschaften:

1. Die Programme dürfen aus beliebig vielen Anweisungen bestehen, die nacheinander als *Sequenz* ausgeführt werden.
2. Wiederholungsanweisungen verschiedener Art, sogenannte *Schleifen*, können, müssen aber nicht verwendet werden.
3. Ausserdem gibt es bedingte Anweisungen, die abhängig vom Programmzustand unterschiedliche Programmausführungen erlauben, die sogenannten *Verzweigungen*.

Enthält eine Programmiersprache Schleifen und Verzweigungen, kann man mit ihrer Hilfe alles berechnen, was überhaupt berechenbar ist. In der Informatik werden solche Programmiersprachen als universell bezeichnet – oder auch als *Turing-vollständig*.

Der Roboter aus dieser Aufgabe ist übrigens eine klassische Umgebung, in der man Programmieren lernen kann. Anstelle des Roboters stellt man sich eine Schildkröte vor, die Linien zeichnet. Solche *Turtle-Graphiken* können beispielsweise mit XLogoOnline aber auch mit TigerJython erstellt werden.

Stichwörter und Webseiten

Turtle-Graphik

- https://de.wikipedia.org/wiki/Strukturierte_Programmierung
- <https://de.wikipedia.org/wiki/Turing-Vollst%C3%A4ndigkeit>
- <https://de.wikipedia.org/wiki/Turtle-Grafik>
- <https://xlogo.inf.ethz.ch/>
- <http://www.tigerjython.ch/>



A. Aufgabenautoren

 Haim Averbuch
 Michelle Barnett
 Michael Barot
 Daniela Bezáková
 Anton Chukhnov
 Allira Crowe
 Andrew Csizmadia
 Christian Datzko
 Susanne Datzko
 Marissa Engels
 Olivier Ens
 Martin Guggisberg
 Vernon Gutierrez
 Juraj Hromkovič
 Alisher Ikramov
 Thomas Ioannou

 Tiberiu Iorgulescu
 Yong-ju Jeon
 Felipe Jiménez
 Anna Laura John
 Mile Jovanov
 Adem Khachnaoui
 Injoo Kim
 Vaidotas Kinčius
 Jia-Ling Koh
 Dennis Komm
 Anja Koron
 Bohdan Kudrenko
 Regula Lacher
 Karolína Mayerová
 Anna Morpurgo
 Tom Naughton

 Pia Niemelä
 Henry Ong
 Wolfgang Pohl
 Stavroula Prantsoudi
 Nol Premasathian
 J.P. Pretti
 Taras Shpot
 Jacqueline Staub
 Bundit Thanasopon
 Monika Tomcsányiová
 Peter Tomcsányi
 Nicole Trachsler
 Troy Vasiga
 Ela Veza
 Florentina Voboril
 Khairul A. Mohamad Zaki



B. Sponsoring: Wettbewerb 2019

HASLERSTIFTUNG

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

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



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

Die RoboRobo Produkte fördern logisches Denken, Vorstellungsvermögen, Fähigkeiten Abläufe und Kombinationen auszudenken und diese systematisch aufzuzeichnen.

Diese Produkte gehören in innovative Schulen und fortschrittliche Familien. Kinder und Jugendliche können in einer Lektion geniale Roboter bauen und programmieren. Die Erwachsenen werden durch die Erfolgserlebnisse der „Erbauer“ miteinbezogen.

RoboRobo ist genial und ermöglicht ein gemeinsames Lern-Erlebnis!



<http://www.baerli-biber.ch/>

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



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



Kanton Zürich
Volkswirtschaftsdirektion
Amt für Wirtschaft und Arbeit

Standortförderung beim Amt für Wirtschaft und Arbeit
Kanton Zürich



i-factory (Verkehrshaus Luzern)

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

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

Wealth Management IT and UBS Switzerland IT



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

bbv Software Services AG ist ein Schweizer Software- und Beratungsunternehmen. Wir stehen für Top-Qualität im Software Engineering und für viel Erfahrung in der Umsetzung. Wir haben uns zum Ziel gesetzt, unsere Expertise in die bedeutendsten Visionen, Projekte und Herausforderungen unserer Kunden einzubringen. Wir sind dabei als Experte oder ganzes Entwicklungsteam im Einsatz und entwickeln individuelle Softwarelösungen.

Im Bereich der Informatik-Nachwuchsförderung engagiert sich die bbv Software Services AG sowohl über Sponsoring als auch über die Ausbildung von Lehrlingen. Wir bieten Schnupperlehrtage an und bilden Informatiklehrlinge in der Richtung Applikationsentwicklung aus. Mehr dazu erfahren Sie auf unserer Website in der Rubrik Nachwuchsförderung.



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

Beratung ist keine Nebensache

Wir interessieren uns, warum, wann und wie die Werbeartikel eingesetzt werden sollen – vor allem aber, wer angesprochen werden soll.



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

OXOcard: Spielend programmieren lernen

OXON



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

Diartis AG

Diartis entwickelt und vertreibt Softwarelösungen für das Fallmanagement.



<https://educatec.ch/>
educaTEC

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



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



AUSBILDUNGS- UND BERATUNGSZENTRUM
FÜR INFORMATIKUNTERRICHT

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

Ausbildungs- und Beratungszentrum für Informatikunterricht der ETH Zürich.



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

Haute école pédagogique du canton de Vaud



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

Pädagogische Hochschule Luzern



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

Pädagogische Hochschule FHNW

Scuola universitaria professionale
della Svizzera italiana



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

La Scuola universitaria professionale della Svizzera italiana (SUPSI)



<https://www.zhdk.ch/>

Zürcher Hochschule der Künste



C. Weiterführende Angebote

Das Lehrmittel zum Informatik-Biber

Module

Verkehr – Optimieren

Musik – Komprimieren

Geheime Botschaften – Verschlüsseln

Internet – Routing

Apps

Auszeichnungssprachen

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

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

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

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

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



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

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



Der Informatik-Biber auf Facebook:

<https://www.facebook.com/informatikbiberch>

010100110101011001001001
010000010010110101010011
010100110100100101000101
001011010101001101010011
010010010100100100100001

SV!A

www.svia-ssie-ssii.ch
schweizerischervereinfürinformatikind
erausbildung//sociétésuissepourl'infor
matique dans l'enseignement//societàsviz
zeraperl'informaticanell'insegnamento

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

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

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