

SOINDEX?



INFORMATIK-BIBER SCHWEIZ  
CASTOR INFORMATIQUE SUISSE  
CASTORO INFORMATICO SVIZZERA



HEILBRONN → H416  
4 6

KANT → K530  
5 3

# Aufgaben und Lösungen 2018

## Schuljahre 7/8



LISSAJOUS → L222  
2 2



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

CASTORO → C236  
3 6 2

LAOYD → L300  
3

Herausgeber:

Christian Datzko, Susanne Datzko, Hanspeter Erni

BIBER → B160  
6 1

GAUSS → G200  
2

A E I O U # W Y	X
B F P V	1
C G J K Q S X Z	2
D T	3
L	4
N M	5
R	6

010100110101011001001001  
010000010010110101010011  
010100110100100101000101  
00101101010101001101010011  
010010010100100100100001

# SV!A

[www.svia-ssie-ssii.ch](http://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



EULER → E460  
6 4

CASTOR → C236  
3 6 2







# Mitarbeit Informatik-Biber 2018

Andrea Adamoli, Christian Datzko, Susanne Datzko, Olivier Ens, Hanspeter Erni, Martin Guggisberg, Carla Monaco, Gabriel Parriaux, Elsa Pellet, Jean-Philippe Pellet, Julien Ragot, Beat Trachler.

Herzlichen Dank an:

Juraj Hromkovič, Urs Hauser, Regula Lacher, Jacqueline Staub: ETHZ

Andrea Maria Schmid, Doris Reck: PH Luzern

Gabriel Thullen: Collège des Colombières

Valentina Dagienė: Bebras.org

Hans-Werner Hein, Ulrich Kiesmüller, Wolfgang Pohl, Kirsten Schlüter, Michael Weigend: Bundesweite Informatikwettbewerbe (BWINF), Deutschland

Chris Roffey: University of Oxford, Vereinigtes Königreich

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

Gerald Futschek, Wilfried Baumann: Oesterreichische Computer Gesellschaft, Österreich

Zsuzsa Pluhár: ELTE Informatikai Kar, Ungarn

Eljakim Schrijvers, Daphne Blokhuis, Arne Heijenga, Dave Oostendorp, Andrea Schrijvers: Eljakim Information Technology bv, Niederlande

Roman Hartmann: hartmannGestaltung (Flyer Informatik-Biber Schweiz)

Christoph Frei: Chragokyberneticks (Logo Informatik-Biber Schweiz)

Andrea Adamoli (Webseite)

Andrea Leu, Maggie Winter, Brigitte Maurer: Senarclens Leu + Partner

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

Die französischsprachige Übersetzung wurde von Nicole Müller und Elsa Pellet und die italienischsprachige Übersetzung von Andrea Adamoli erstellt.



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

Der Informatik-Biber 2018 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 2018 geprüft. Dieses Aufgabenheft wurde am 9. Oktober 2019 mit dem Textsatzsystem  $\text{\LaTeX}$  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. 39 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 2018 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 („Kleiner Biber“: 108 Punkte, Stufen 5 und 6: 144 Punkte) 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

Hanspeter Erni

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

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

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



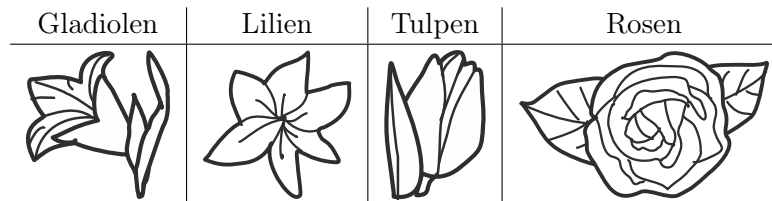
# Inhaltsverzeichnis

Mitarbeit Informatik-Biber 2018	i
Vorwort	ii
1. Claras Blumen	1
2. Liniennetz	3
3. Planet Z	5
4. Gelateria	7
5. Das Pfeil-Labyrinth	9
6. Ausflug mit Aussicht	11
7. Lügen haben kurze Beine	13
8. Wasserfälle	17
9. Biber-Teich	21
10. Biber-Wettbewerb	23
11. Ferienhaus Nr. 29	25
12. Nachbarn	27
13. Computerspiel	31
14. Biberbesuch	33
15. Zwei Biber bei der Arbeit	37
A. Aufgabenautoren	39
B. Sponsoring: Wettbewerb 2018	40
C. Weiterführende Angebote	43



# 1. Claras Blumen

Clara mag bunte Blumensträuße und besucht daher einen Blumenladen. Da sind die folgenden Blumenarten zu finden:

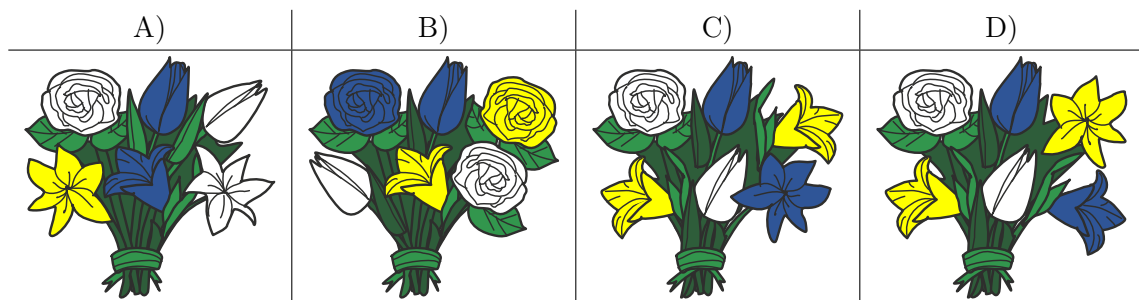


Jede Blumenart ist in den folgenden Farben erhältlich: Weiss, **Blau** und **Gelb**.

Clara möchte einen Blumenstrauß mit sechs Blumen, der die folgenden Bedingungen erfüllt:

1. Jede Farbe Weiss, Blau und Gelb soll genau zweimal vorkommen.
2. Blumen der gleichen Art sollen nicht die gleiche Farbe haben.
3. Jede Blumenart soll höchstens zweimal vorkommen.

Welcher Blumenstrauß erfüllt alle drei Bedingungen?





## Lösung

Die richtige Antwort ist D). Im Blumenstrauß A) gibt es drei weiße Blüten (Regel 1 ist verletzt), in B) drei Rosen (Regel 3 ist verletzt), und im Blumenstrauß C) haben zwei Blüten derselben Blumenart die gleiche Farbe (Regel 2 ist verletzt).

## Dies ist Informatik!

Allgemeine Informatikprobleme werden durch eine Reihe von Einschränkungen beschrieben. Die Aufgabe besteht darin, eine Lösung zu finden, die all diese Einschränkungen oder so viele wie möglich erfüllt.

In der Informatik hat man häufig komplexere Aufgaben, bei denen die Beschränkungen beispielsweise durch logische Operatoren wie die UND-Verknüpfung ( $A$  und  $B$  bedeutet, dass die beiden Bedingungen  $A$  und  $B$  gleichzeitig erfüllt werden müssen, wie die drei Regeln in unserer Aufgabe) oder die ODER-Verknüpfung ( $A$  oder  $B$  bedeutet, dass es genügt, wenn der Bedingungen erfüllt wird) gegeben sind.

## Stichwörter und Webseiten

Bedingungen, logische Operatoren

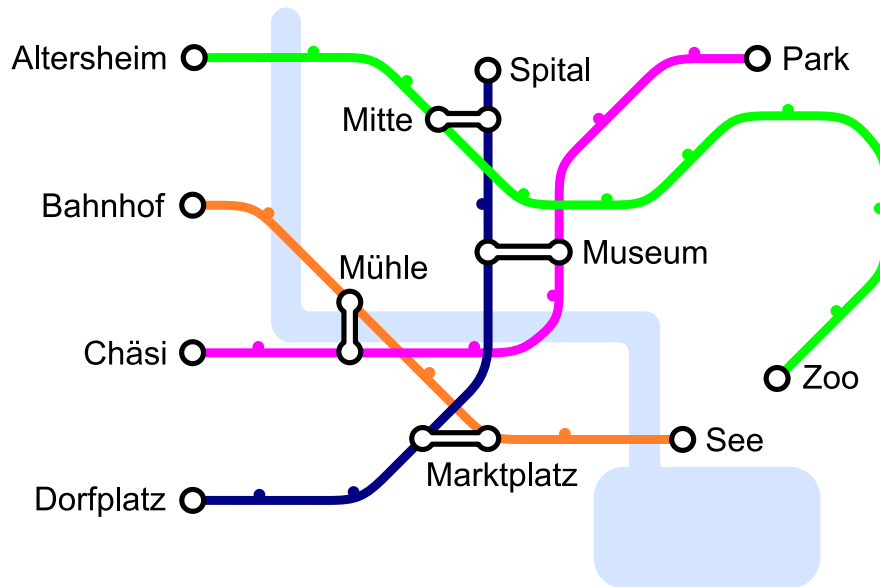
- <https://bookofbadarguments.com/de/?view=allpages>
- [https://de.wikipedia.org/wiki/Boolesche\\_Algebra](https://de.wikipedia.org/wiki/Boolesche_Algebra)
- <https://www.iep.utm.edu/prop-log/>





## 2. Liniennetz

Es gibt vier Linien, die an den Stationen „Altersheim“, „Bahnhof“, „Chäsi“ und „Dorfplatz“ starten. Es gibt vier Kreuzungsstationen „Museum“, „Marktplatz“, „Mühle“ und „Mitte“, an denen man von einer Linie auf die andere wechseln kann.



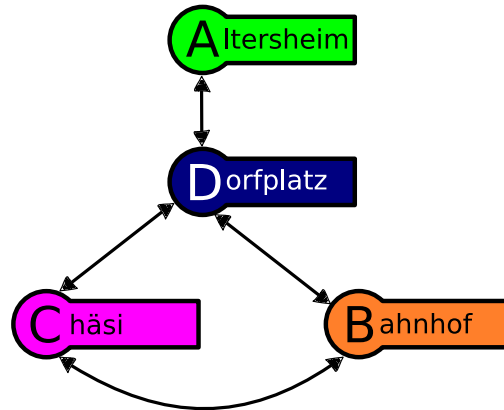
*Johannes möchte zum Zoo. Er weiss, dass er nur einmal die Linie wechseln muss. Bei welcher Station startet seine Linie?*



## Lösung

Die richtige Antwort ist die Station „Dorfplatz“. Wenn man vom Zoo rückwärts fährt, gibt es nur eine Kreuzungsstation („Mitte“), an der man auf die Linie wechselt, die an der Station „Dorfplatz“ startet.

Man kann das Liniennetz auch mit Hilfe eines Graphen darstellen, der darlegt, von welcher Linie man mit einmal Umsteigen auf welche andere Linie kommt:



Von	Nach		
Altersheim ↔ Zoo	Dorfplatz ↔ Spital		
Bahnhof ↔ See	Chäsi ↔ Park	Dorfplatz ↔ Spital	
Chäsi ↔ Park	Bahnhof ↔ See	Dorfplatz ↔ Spital	
Dorfplatz ↔ Spital	Bahnhof ↔ See	Chäsi ↔ Park	Altersheim ↔ Zoo

Wenn man also mit einmal Umsteigen auf die Linie „Altersheim ↔ Zoo“ möchte, kann man das nur von der Linie „Dorfplatz ↔ Spital“ von der Station „Dorfplatz“ her kommend, denn nur von der erreicht man die Linie „Altersheim ↔ Zoo“ mit einem Mal Umsteigen.

## Dies ist Informatik!

Kommt Dir das Liniennetz irgendwie bekannt vor? Richtig, viele Liniennetze von Bussen, Trams oder Metros sehen so aus. Das ist kein Zufall sondern eine Erfindung von Harry Beck, der 1931 einen solchen Plan für das Londoner U-Bahn-System erfand.

In der Informatik nennt man einen solchen abstrakten Plan einen Graph, der aus Knoten (die Stationen) und Kanten (die Strecke zwischen zwei Stationen) besteht. In unserem Fall wird noch zwischen Knoten, die nur eine oder zwei Kanten haben (Endstationen sowie reguläre Stationen), und Knoten die mehrere Kanten haben (Kreuzungsstationen) unterschieden.

Graphen werden auch noch an vielen anderen Orten verwendet. Beziehungen von Menschen in sozialen Netzwerken, Routenplaner oder auch Shopping-Vorschläge werden mit Graphen modelliert. Es ist also eine wichtige Informatik-Kompetenz, mit Graphen umgehen zu können.

## Stichwörter und Webseiten

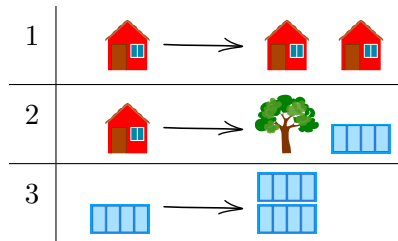
Liniennetz, Graph

- [https://de.wikipedia.org/wiki/Tube\\_map](https://de.wikipedia.org/wiki/Tube_map)
- [https://de.wikipedia.org/wiki/Graph\\_\(Graphentheorie\)](https://de.wikipedia.org/wiki/Graph_(Graphentheorie))



### 3. Planet Z

Die Bewohner des Planeten Z bauen ihre Städte immer auf die gleiche Weise. Sie beginnen eine Stadt mit einem Haus. Dann ersetzen Sie einzelne Objekte nach folgenden drei Regeln:



Wenn man zuerst Regel 1, dann Regel 2 und dann zwei Mal Regel 3 anwendet, erhält man beispielsweise die Stadt ganz rechts im Bild:



Beachte, dass die Reihenfolge der einzelnen Objekte nicht verändert werden kann.  
 Welche der folgenden Städte kann nicht auf dem Planeten Z stehen?





## Lösung

Die richtige Antwort ist B). Bäume können nur durch Regel 2 in eine Stadt gebracht werden. Regel 2 besagt aber, dass rechts neben einem Baum ein Block stehen muss. In Stadt B) ist rechts neben dem rechten Baum kein Block. Es gibt auch keine Regel, mit der man Blöcke entfernen kann. Deshalb kann die Stadt B nicht auf dem Planeten Z stehen.

Stadt A) kann man durch Anwendung der folgenden Regeln bauen: 1, 2, 3 und dann wieder 3.

Stadt C) kann man bauen, indem man Regel 1 drei Mal anwendet.

Stadt D) kann man so bauen: Zuerst wendet man Regel 1 an. Anschliessend wendet man Regel 2 auf jedes Haus an und danach wendet man Regel 3 auf jeden Block zweimal an.

## Dies ist Informatik!

Die Regeln aus der Aufgabe nennt man Ersetzungsregeln: Ein Symbol oder ein Objekt wird durch eine Folge von anderen Symbolen oder Objekten ersetzt. Wenn immer nur ein Symbol oder Objekt ersetzt wird, nennt man ein solches Regelsystem kontextfrei. Ein Symbol oder Objekt wird durch etwas anderes ersetzt, ohne den Kontext des Symbols oder Objekts (also das, was rechts und links von dem Symbol oder Objekt ist) zu beachten.

Ersetzungsregeln verwendet man in der Informatik beispielsweise, um die Syntax einer Programmiersprache zu definieren. Die Symbole oder Objekte sind Schlüsselbegriffe und die Regeln beschreiben, wie man sie zu einem (syntaktisch) korrekten Programm zusammensetzt. In der Aufgabe sind die Symbole oder Objekte die Häuser, Bäume und Blöcke. Die Symbole und Objekte zusammen mit den Ersetzungsregeln bilden dann eine Grammatik, die eine Sprache beschreiben.

Wenn der Computer ein Programm in Maschinsprache übersetzt (compiliert) oder direkt ausführt (sogenannte „interpretierte“ Programme, häufig auch Skripte genannt), prüft er zunächst, ob der Programmtext tatsächlich den Regeln der Programmiersprache folgt. Er versucht also, mit Hilfe eines Syntaxbaumes die Ersetzungsregeln nachzubauen, die aus dem Startsymbol (in der Aufgabe ein Haus) den Programmtext (in der Aufgabe die vier verschiedenen Antwortmöglichkeiten) erzeugen. Das ist in unserem Fall einfach möglich, da *Wörter* (Abfolgen von Symbolen oder Objekten, in Fall dieser Aufgabe eine Stadt) immer grösser werden, aber nie kleiner.


## Stichwörter und Webseiten

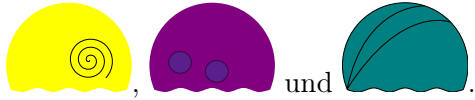
Ersetzungsregel, Grammatik, kontextfreie Sprachen

- <https://de.wikipedia.org/wiki/Produktionsregel>
- [https://de.wikipedia.org/wiki/Kontextfreie\\_Sprache](https://de.wikipedia.org/wiki/Kontextfreie_Sprache)
- <https://de.wikipedia.org/wiki/Syntaxbaum>



## 4. Gelateria

Beim Dorfbrunnen gibt es zwei Gelaterias. Sie bieten dieselben vier Glace-Sorten an: ,

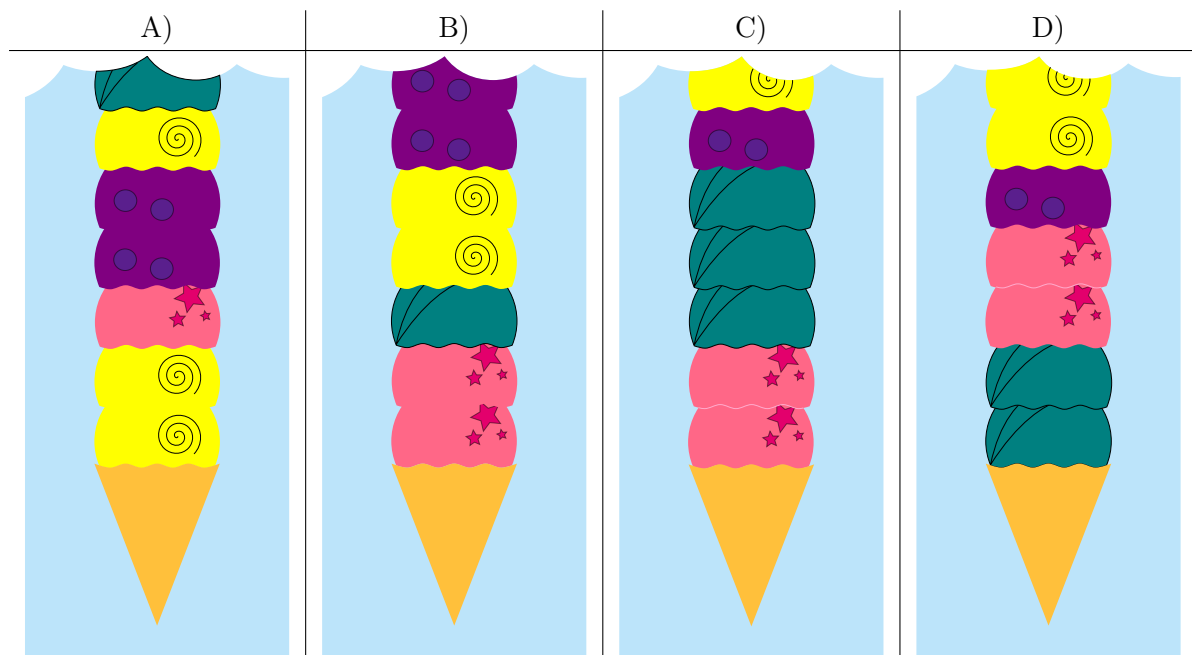


Die erste Gelateria verwendet die folgenden Anweisungen für ein Cornet:

1. Nimm ein leeres Cornet.
2. Wähle eine zufällige Glace-Sorte und gib zwei Kugeln dieser Glace-Sorte hinzu.
3. Gib eine Kugel mit einer der drei anderen Glace-Sorten hinzu.
4. Wenn die gewünschte Kugelzahl erreicht ist, höre auf. Ansonsten mache mit Schritt 2 weiter.

Die zweite Gelateria folgt überhaupt keinen Anweisungen.

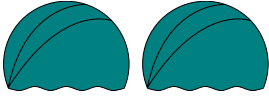
Auf dem Bild siehst du die unteren Kugeln einiger Cornets. Welches Cornet stammt mit Sicherheit aus der zweiten Gelateria?







## Lösung

Die richtige Antwort ist D). Das ist das einzige Cornet, das ganz eindeutig nicht nach den Anweisungen hergestellt worden ist. Es beginnt zwar korrekt mit zwei Kugeln der gleichen Glace-Sorte



gefolgt von einer Kugel einer anderen Glace-Sorte



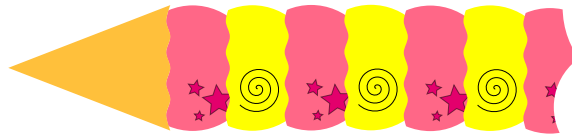
kommen zwei Kugeln unterschiedlicher Glace-Sorten  , obwohl nach den Anweisungen eigentlich wieder zwei gleiche Glace-Sorten kommen müssten.

Die Antworten A), B), und C) sind nicht korrekt. Alle diese Cornets entsprechen, soweit man es erkennen kann, den Anweisungen.



## Dies ist Informatik!

Muster in Cornets, Texten oder Bildern können durch Anweisungsfolgen erzeugt werden. Informatikerinnen und Informatiker entwickeln Computerprogramme, mit denen Muster oder Abweichungen von Mustern erkannt werden können. Manchmal entstehen Muster durch Wiederholung von Anord-

nungen. Dieses Muster



ist beispielsweise ein einfaches

Muster, das durch Wiederholung von  und  entstanden ist. Solche Muster sind leicht zu erkennen. In der Aufgabe ist die Sache schwieriger, weil die Anweisungen der ersten Gelateria auch Zufallsentscheidungen enthalten.

Im allgemeinen Fall kann man niemals ganz sicher sein, ob eine Reihenfolge durch Zufall oder durch eine Folge von Anweisungen erzeugt worden ist. Bei den Beispielen dieser Aufgabe konnten wir nur bei einem Cornet sagen, dass es sicher nicht den Anweisungen entsprach und deshalb aus der zweiten Gelateria stammen musste. Man kann aber aufgrund der Zusammensetzung des Cornets niemals sicher entscheiden, ob es aus der ersten Gelateria stammt, denn es könnte ja auch zufälligerweise zu den Anweisungen passen.

## Stichwörter und Webseiten

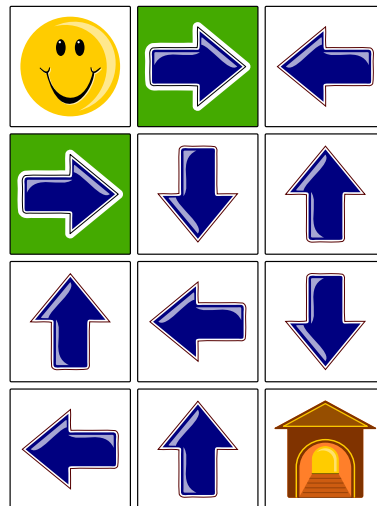
Mustererkennung

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



## 5. Das Pfeil-Labyrinth

Das Smiley 😊 möchte nach Hause 🏠. Aber dazu muss es durch das Pfeil-Labyrinth. Es kann einen der beiden Eingänge (grüne Felder) benutzen. Wenn es auf einem Feld mit einem Pfeil steht, muss es das Feld in Richtung des Pfeiles verlassen. So wie die Pfeile momentan ausgerichtet sind, kann es unmöglich nach Hause finden.

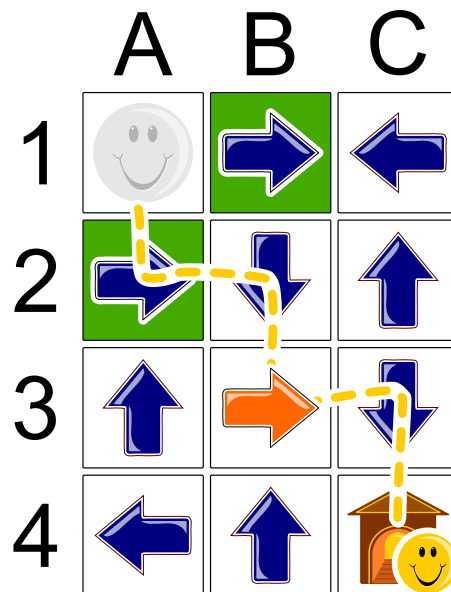


*Welchen Pfeil muss man ändern, so dass das Smiley nach Hause kommt?*



## Lösung

Der Pfeil, dessen Richtung geändert wurde, ist rot markiert. Der Weg des Smiley nach Hause ist zu erkennen: A1 – A2 – B2 – B3 – C3 – C4.



Man kann sogar beweisen, dass dies die einzig mögliche Lösung ist. Wenn man am Ziel C4 beginnt und rückwärts geht, kann man aus zwei Richtungen zum Ziel C4 gelangen: Von C3 und von B4. Der Pfeil in B4 zeigt aber in die falsche Richtung und müsste geändert werden. Weil aber kein Nachbarfeld nach B4 zeigt, müsste ein zweiter Pfeil geändert werden, was nicht erlaubt ist. Folglich ist das Ziel nur über das andere Nachbarfeld C3 erreichbar. Der Pfeil in diesem Feld steht ja auch schon richtig und zeigt auf das Ziel. Nun gibt es in den Nachbarfeldern von C3 keinen Pfeil, der auf C3 zeigt. Einer der beiden Pfeile müsste also geändert werden, B3 oder C2. Nun gibt es aber keinen Pfeil, der auf C2 zeigt. C3 ist also von einem Eingang nicht zu erreichen, ohne dass ein weiterer Pfeil geändert wird. Deshalb muss der Weg über B3 gehen. B3 ist von einem Eingang erreichbar (A1 – A2 – B2), ohne dass weitere Pfeile geändert werden müssten. Das ist damit die einzige Möglichkeit.

## Dies ist Informatik!

Bei dieser Aufgabe wurde ein Weg durch das Pfeil-Labyrinth gesucht, bei dem nur ein Pfeil geändert werden musste. Wie findet man eigentlich die Lösung zu solch einem Problem? Wie könnte das ein Computer machen? Bei der Beweisführung im letzten Abschnitt wurde eine Vorgehensweise verwendet, die man *backtracking* nennt. Das bedeutet in etwa: Man verfolgt eine Spur (engl. *track*) bis es nicht mehr weitergeht (egal ob vom Start oder vom Ziel aus). Dann nimmt man den letzten Schritt zurück und prüft von dieser Stelle aus eine andere Spur. Unmögliche Wege kann man dabei von vornherein ausschliessen und findet garantiert eine Lösung, wenn es sie gibt, weil man ja alle Möglichkeiten ausprobiert.

## Stichwörter und Webseiten

Labyrinth, Backtracking

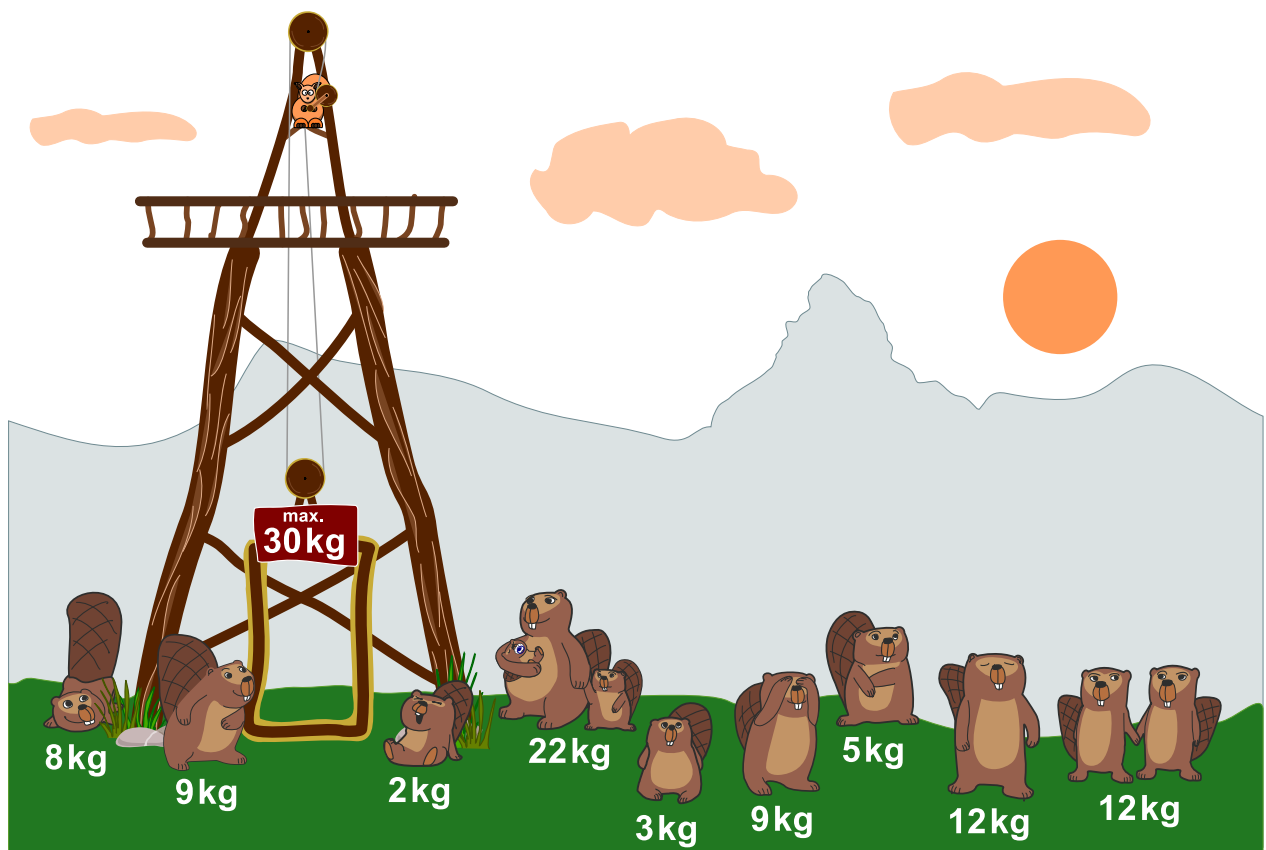
- <https://de.wikipedia.org/wiki/Backtracking>





## 6. Ausflug mit Aussicht

Eine Biberfamilie macht einen Ausflug zu einem Aussichtsturm. Sie sind spät dran. Der Lift fährt nur noch zweimal hoch. Der Lift darf nicht mit mehr als 30 kg pro Mal beladen werden. Die Zwillinge gehen nur gemeinsam auf den Turm. Die Bibermama hält das Baby im Arm und ein Biberkind an der Hand. Es sollen aber möglichst viele Biber auf den Aussichtsturm.



*Es muss schnell entschieden werden und nur die folgenden fünf Optionen sind möglich. Wer muss unten bleiben, damit möglichst viele Biber auf den Aussichtsturm können?*

- A) Alle können mitfahren.
- B) Die Bibermama mit dem Baby und dem Biberkind.
- C) Die Zwillinge und der 5 kg-Biber.
- D) Die Zwillinge und die Bibermama mit dem Baby und dem Biberkind.
- E) Die Bibermama mit dem Baby und dem Biberkind und der 12 kg-Biber.

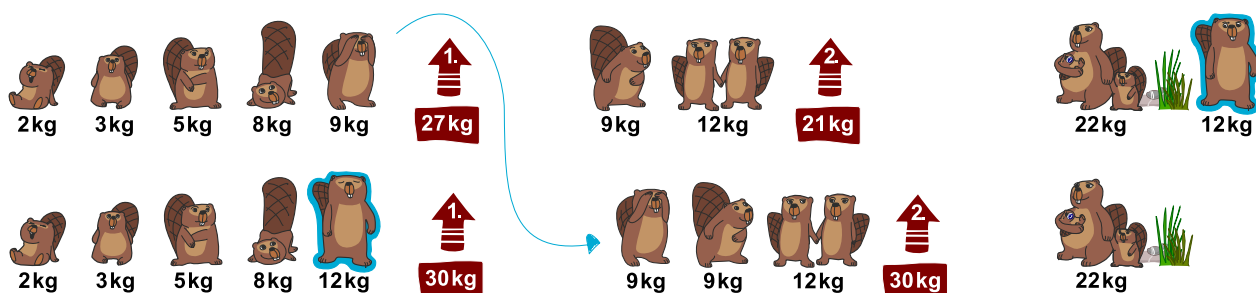


## Lösung

Eine richtige Antwort ist B):



Ein Lösungsweg könnte sein, die Kabinen mit möglichst vielen von den leichten Bibern zu beladen:  $2\text{ kg} + 3\text{ kg} + 5\text{ kg} + 8\text{ kg} + 9\text{ kg} = 27\text{ kg}$  in die erste Kabine und in die zweite  $9\text{ kg} + 12\text{ kg} = 21\text{ kg}$  (das sind 8 Biber). Aber es passt noch ein weiterer Biber in den Lift:



Mit dieser Strategie können die beiden Fahrten optimal genutzt werden: Der 9 kg schwere Biber in der ersten Kabine wird durch den 12 kg schweren Biber ersetzt. Das Maximalgewicht von 30 kg ist erreicht. Der 9 kg schwere Biber kann in die zweite Kabine einsteigen, die dann ebenfalls ihr Maximalgewicht von 30 kg erreicht hat.

Die anderen Lösungen sind nicht möglich, weil sie entweder das Maximalgewicht überschreiten (alle Biber zusammen sind mehr als 60 kg und selbst wenn die Zwillinge und der 5 kg-Biber nicht dabei sind, sind es noch 65 kg), oder weil sie schlechter sind (wenn die Mutter mit den Kindern dabei ist, können die Zwillinge oder der 12 kg-Biber mitfahren).

## Dies ist Informatik!

Die optimale Kombination als Lösung für ein Problem zu finden ist eines der klassischen Probleme der Informatik. Häufig lassen sie sich nicht schnell genug oder gar nicht innert nützlicher Zeit lösen, da es zu viele mögliche Lösungen gibt, die überprüft werden müssten. Die Informatik nennt so etwas ein praktisch unlösbares Problem.

Dieses Problem nennt man ein *Rucksack-Problem*, bei dem möglichst viele Objekte eingepackt werden müssen ohne ein Maximalgewicht zu überschreiten. Dieses und vergleichbare Probleme heissen *NP-vollständig*. Sie können nur näherungsweise gelöst werden, das bedeutet, dass man eine mögliche und gute Lösung finden kann, aber nicht notwendigerweise die optimale Lösung. Dies erreicht man, indem man sich eine sinnvolle Strategie („Heuristik“) überlegt.

## Stichwörter und Webseiten

Optimierung, Rucksack-Problem

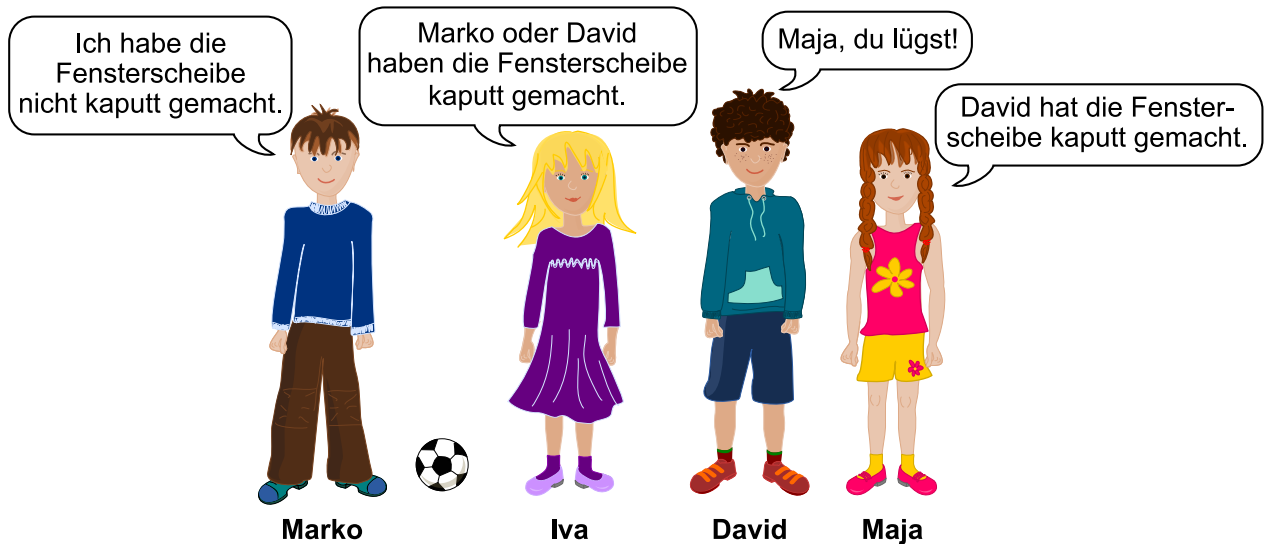
- [https://de.wikipedia.org/wiki/Kombinatorische\\_Optimierung](https://de.wikipedia.org/wiki/Kombinatorische_Optimierung)
- <https://de.wikipedia.org/wiki/Rucksackproblem>



## 7. Lügen haben kurze Beine

Eines schönen Tages spielen die Kinder Maja, David, Iva und Marko Fussball in der Nähe von Anna's Haus. Auf einmal zerbricht eine Fensterscheibe und Anna möchte wissen, wer es war. Anna kennt die vier Kinder und weiss, dass drei von ihnen immer die Wahrheit sagen, bei dem vierten Kind weiss sie es nicht.

Die vier Kinder sagen:

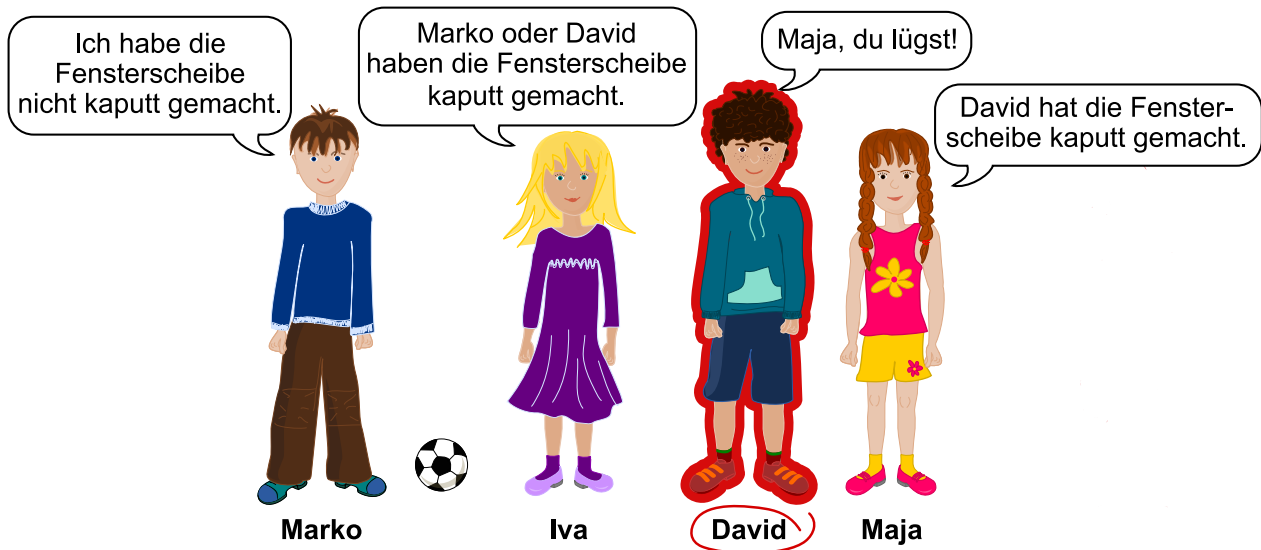


*Welches Kind hat die Fensterscheibe kaputt gemacht?*



## Lösung

David hat die Fensterscheibe kaputt gemacht.



Die Aussagen von Maja und David können nicht beide wahr sein, einer von den beiden muss also lügen. Wenn Maja die Wahrheit sagen würde, würde also David lügen. Die Aussagen von Iva und Marko könnten ebenso korrekt sein. Wenn David aber die Wahrheit sagen würde, müsste also Maja lügen; zusätzlich müsste aber auch Iva oder Marko lügen! Das wäre aber nicht möglich, da drei Kinder immer die Wahrheit sagen.

## Dies ist Informatik!

Um diese Aufgabe zu lösen musst du logisch denken. Die zugrunde liegende Logik wurde 1854 von George Boole (1815 – 1864) formuliert, der logische Aussagen auf ihre Grundlagen zurückgeführt hat und beschrieben hat.

Nach ihm ist eine Aussage entweder *wahr* oder *falsch* (Prinzip *tertium non datur*). Aussagen können mit Hilfe von *Operatoren* kombiniert werden. Einfache logische Operatoren wie *UND* oder *ODER* verbinden zwei Aussagen zu einer neuen Aussage. Es gibt auch Operatoren (wie das *NICHT*), die lediglich eine Aussage nehmen und sie verändern. Wann so kombinierte Aussagen letztlich wahr sind, kann man mit Hilfe von *Wahrheitstabellen* einfach herausfinden.

Einer der Operatoren, die das Schliessen „WENN“ → „DANN“ modelliert, ist die Subjunktion. Man spricht dann davon, dass man „logische Schlüsse zieht“. Diese wird zum Lösen dieser Aufgabe benötigt.

Computer basieren ebenfalls auf boole'schen Aussagen und einfachen logischen Operatoren, da man diese sehr einfach und in grosser Stückzahl bauen kann. Es gibt zwar auch einige Computer, die auf anderen Systemen basieren (beispielsweise ternäre Computer aus den späten 50er Jahren in Russland), diese sind aber entweder nur experimentell geblieben oder haben nie grosse Stückzahlen erreicht.

## Stichwörter und Webseiten

Logisches Schliessen

- [https://de.wikipedia.org/wiki/Satz\\_vom\\_ausgeschlossenen\\_Dritten](https://de.wikipedia.org/wiki/Satz_vom_ausgeschlossenen_Dritten)




- [https://de.wikipedia.org/wiki/George\\_Boole](https://de.wikipedia.org/wiki/George_Boole)
- [https://de.wikipedia.org/wiki/Boolesche\\_Algebra](https://de.wikipedia.org/wiki/Boolesche_Algebra)
- <https://de.wikipedia.org/wiki/Subjunktion>
- [https://de.wikipedia.org/wiki/Ternärer\\_Computer](https://de.wikipedia.org/wiki/Ternärer_Computer)

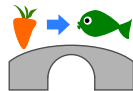




## 8. Wasserfälle

 Katja sitzt oben auf dem Berg. Der Berg hat drei Wasserfälle. Die Wasserfälle fließen in einen Fluss.

Katja kann entweder ein Rübli oder einen Fisch in einen der Wasserfälle fallen lassen. Der Fluss hat mehrere Brücken mit Trolle. Die Trolle ersetzen Gegenstände, die unter den Brücken durchgehen.

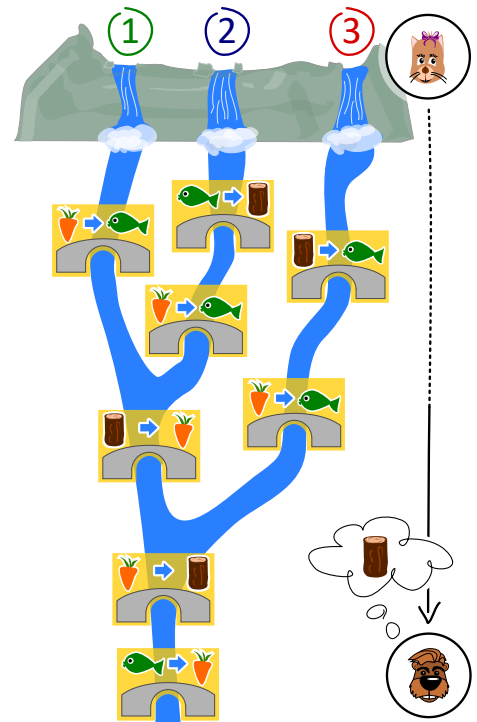


Wenn zum Beispiel ein Rübli unter einer solchen Brücke durchschwimmt, ersetzen die Trolle das Rübli durch einen Fisch.

 Justus sitzt am Ende des Flusses.

*Justus braucht Holz. Welchen Gegenstand muss Katja wählen und in welchen Wasserfall muss sie ihn fallen lassen, damit Justus Holz bekommt?*

- A) Sie lässt einen Fisch 🐟 in den Wasserfall 1 fallen.
- B) Sie lässt einen Fisch 🐟 in den Wasserfall 2 fallen.
- C) Sie lässt ein Rübli 🥕 in den Wasserfall 2 fallen.
- D) Sie lässt ein Rübli 🥕 in den Wasserfall 3 fallen.





## Lösung

Die richtige Antwort ist: B) Sie lässt einen Fisch 🐟 in den Wasserfall 2 fallen.

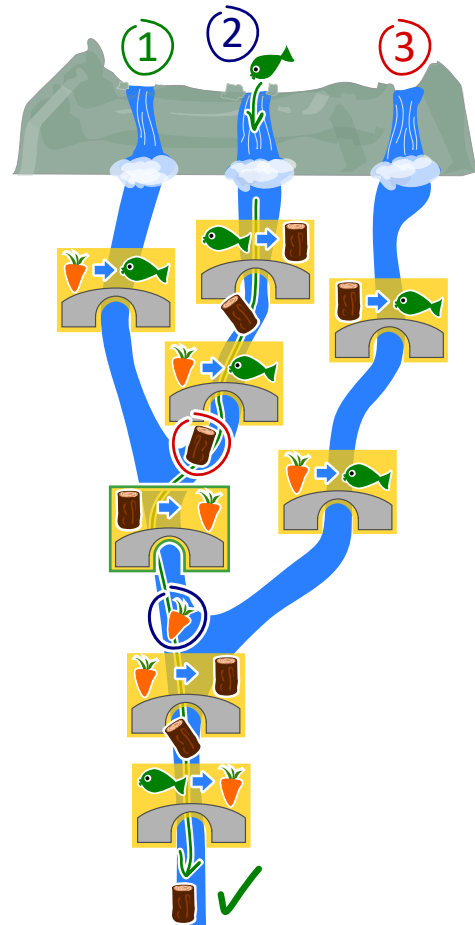
Was passiert bei den verschiedenen Lösungsvorschlägen:

- A) Ein in den Wasserfall 1 geworfener Fisch wird nur unter der letzten Brücke gewechselt. So erhält Justus ein Rüeblli.
- B) Ein in den Wasserfall 2 geworfener Fisch wird in Holz, dann Rüeblli und dann wieder in Holz umgewandelt. So erhält Justus Holz. Das ist somit korrekt.
- C) Ein Rüeblli, die in den Wasserfall 2 fallen gelassen wird, wird in Fisch und dann Rüeblli geändert. So bekommt Justus ein Rüeblli.
- D) Ein Rüeblli, die in den Wasserfall 3 fällt, wird in Fisch und dann in Rüeblli verwandelt. So bekommt Justus ein Rüeblli.

Ein anderer Lösungsansatz für diese Aufgabe besteht darin, rückwärts zu beginnen:

Um am Ende Holz zu bekommen, muss der gefallene Gegenstand ein Rüeblli sein, wenn er die vorletzte Brücke passiert.

Die einzige Möglichkeit, an diesem Punkt ein Rüeblli 🥕 zu haben, ist, wenn der Gegenstand die einzige gemeinsame Brücke passiert hat, die Wasserfall 1 und 2 gemeinsam haben (und nicht 3). Die einzige von den vier Antwortmöglichkeiten, an diesem Punkt Holz 🍷 zu haben, ist, einen Fisch in den Wasserfall 2 fallen zu lassen.



## Dies ist Informatik!

Man kann sich einen Computer als ein Gerät vorstellen, das Eingaben liest, diese verarbeitet und Ausgaben schreibt. Wie „weiss“ ein Computer aber, was zu tun ist? Die Antwort ist, dass Menschen ihm vorher befohlen haben, was zu tun ist. Das macht man, indem man Programme schreibt. Das Analysieren von Programmen nennt man Testen von Software.

Es gibt viele verschiedene Programmiersprachen, die nach unterschiedlichen Programmierparadigmen funktionieren. Ein solches Programmierparadigma ist die funktionale Programmierung. Dieser Programmierstil ist wie ein kleiner Computer, da er aus vielen Funktionen besteht, die eine Eingabe verarbeiten und daraus eine Ausgabe erzeugen. Die Brücken in dieser Aufgabe sind wie kleine Funktionen und das vollständige System ist wie ein Programm, das mit einer funktionalen Programmiersprache geschrieben wurde.

## Stichwörter und Webseiten

Test von Software, Programmierparadigma, Funktionale Programmierung, Funktionen und Parameter





- <https://de.wikipedia.org/wiki/Softwaretest>
- <https://de.wikipedia.org/wiki/White-Box-Test>
- <https://de.wikipedia.org/wiki/Black-Box-Test>
- <https://de.wikipedia.org/wiki/Programmierparadigma>
- [https://de.wikipedia.org/wiki/Funktionale\\_Programmierung](https://de.wikipedia.org/wiki/Funktionale_Programmierung)

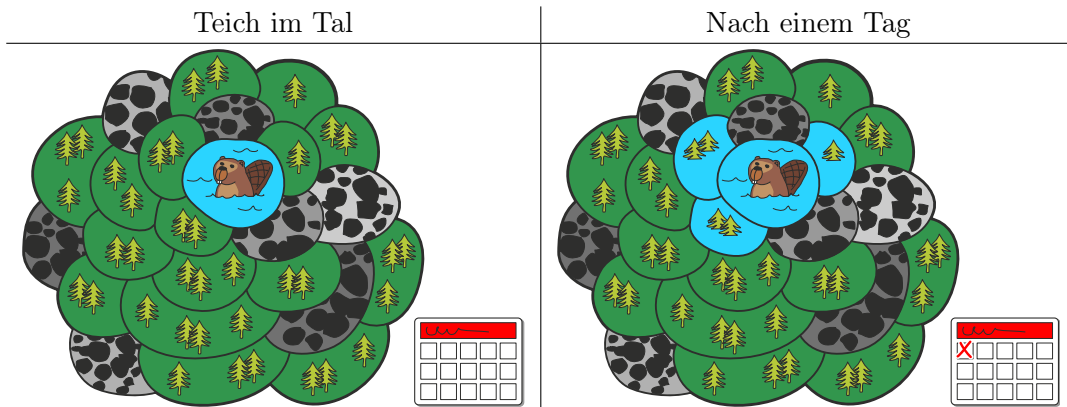




## 9. Biber-Teich

In einem Tal liegt ein kleiner Teich. Er ist umgeben von Landstücken mit Wald oder mit Felsen. Im Teich leben einige Biber.

Eines Tages wird den Bibern der Teich zu klein und nun fluten sie den Wald. An jedem Tag fluten sie alle Waldstücke, die an bereits geflutete Waldstücke angrenzen. Nach einem Tag sind drei Waldstücke geflutet.



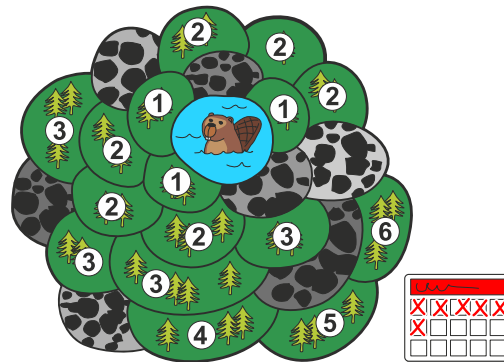
*Nach wie vielen Tagen insgesamt (also inklusive dem dargestellten ersten Tag) sind alle Waldstücke geflutet?*



## Lösung

Nach sechs Tagen sind alle Waldstücke geflutet.

Das Bild zeigt für jedes Waldstück, am wievielten Tag es geflutet wird: Die Waldstücke, die an den See angrenzen, sind nach einem Tag geflutet und deshalb mit der Zahl 1 markiert. Die Waldstücke, die an diese Felder angrenzen, sind mit der Zahl 2 markiert; sie sind nach zwei Tagen geflutet, und so weiter. Am sechsten Tag wird ein letztes Waldstück auf diese Weise markiert. Nach sechs Tagen ist also dieses Waldstück geflutet – und damit ist das ganze Tal geflutet.



## Dies ist Informatik!

In dieser Biberaufgabe fluten die Biber ein zusammenhängendes Waldgebiet, das neben dem ursprünglichen Teich aus einzelnen benachbarten Waldstücken besteht. Das Gebiet ist zusammenhängend, weil man von jedem Waldstück in dem Gebiet über andere Waldstücke zu jedem anderen Waldstück gehen kann.

Auch ausserhalb des Tals mit dem Teich der Biber gibt es zusammenhängende Gebiete, die geflutet werden müssen. Ein einfarbiger Bereich in einem Bild ist letztlich ein zusammenhängendes Gebiet gleichfarbiger Pixel. Eine Gruppe von Jugendlichen, in denen jeder über beliebig viele andere Jugendliche mit jedem anderen Jugendlichen der Gruppe befreundet ist, ist auch ein „zusammenhängendes Gebiet“, wenn man die direkte Freundschaftsbeziehung zwischen zwei Jugendlichen als Nachbarschaft betrachtet.

Die Informatik kennt Methoden, zusammenhängende Gebiete zu ermitteln, etwa die Breitensuche oder die Tiefensuche. Mit Hilfe dieser Methoden können beispielsweise Bereiche in Bildern umgefärbt oder Gruppierungen in sozialen Netzwerken ermittelt werden.

## Stichwörter und Webseiten

Wavefront Algorithm, Breadth-First Search

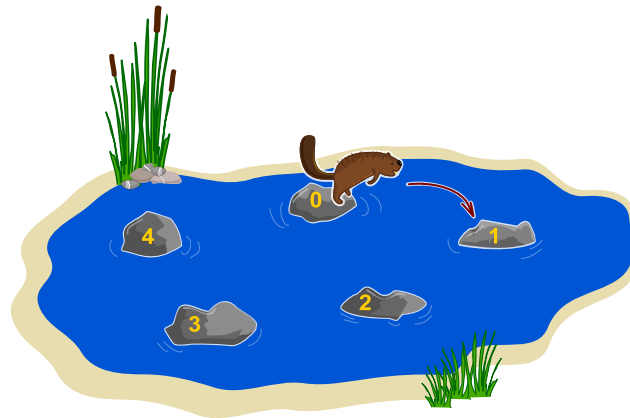
- [https://de.wikipedia.org/wiki/Zusammenhang\\_\(Graphentheorie\)](https://de.wikipedia.org/wiki/Zusammenhang_(Graphentheorie))
- <https://de.wikipedia.org/wiki/Breitensuche>



## 10. Biber-Wettbewerb

Als Vorbereitung für den alljährlichen Biber-Wettbewerb trainieren einige Biber intensiv. Die heutige Trainingseinheit besteht darin, im Uhrzeigersinn von Fels zu Fels zu springen, wie es der Pfeil zeigt. Wenn der Biber 8 mal springt, landet er am Ende auf Fels Nummer 3:

$0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 0 \rightarrow 1 \rightarrow 2 \rightarrow 3.$



*Der stärkste Biber springt heute ganze 129 mal. Er startet auf Fels 0. Auf welchem Fels ist er am Ende gelandet?*



## Lösung

Wenn ein Biber 5 mal springt, landet er auf dem Fels von dem er gestartet ist. Wir nennen das eine „Runde“. Um herauszufinden, wo er nach 129 Sprüngen landet, müssen wir herausfinden, wie viele Runden er springt und wie viele Sprünge er danach noch vor sich hat. In diesem Fall sind es  $129 = 25 \cdot 5 + 4$  Sprünge (25 Runden plus 4 Sprünge). Wenn er also 129 mal springt, endet er auf demselben Fels wie wenn er 4 mal gesprungen wäre. Daher ist die richtige Antwort 4.

## Dies ist Informatik!

Du hast so etwas vielleicht schon im Mathematikunterricht kennen gelernt. Es ist dasselbe wie *der Rest der ganzzahligen Division*, manchmal auch schriftliches Teilen oder Euklidisches Teilen genannt. In dieser Aufgabe muss also der Rest der ganzzahligen Division von  $129 : 5 = 25$  Rest 4 ermittelt werden. Da dies in Computern sehr häufig berechnet werden muss, gibt es alleine für das Berechnen des Rests einen eigenen Namen: es ist die Modulo-Operation. Normalerweise wird das Zeichen „%“ oder auch „mod“ als Operator verwendet. Man kann also schreiben:  $129 \% 5 = 4$ .

Diese Operation wird beispielsweise in Schleifen (so wie der Biber, der in Runden springt), wenn Variablen überlaufen, oder auch für das weit verbreitete kryptographische Verfahren RSA verwendet.

## Stichwörter und Webseiten

Modulo-Operation

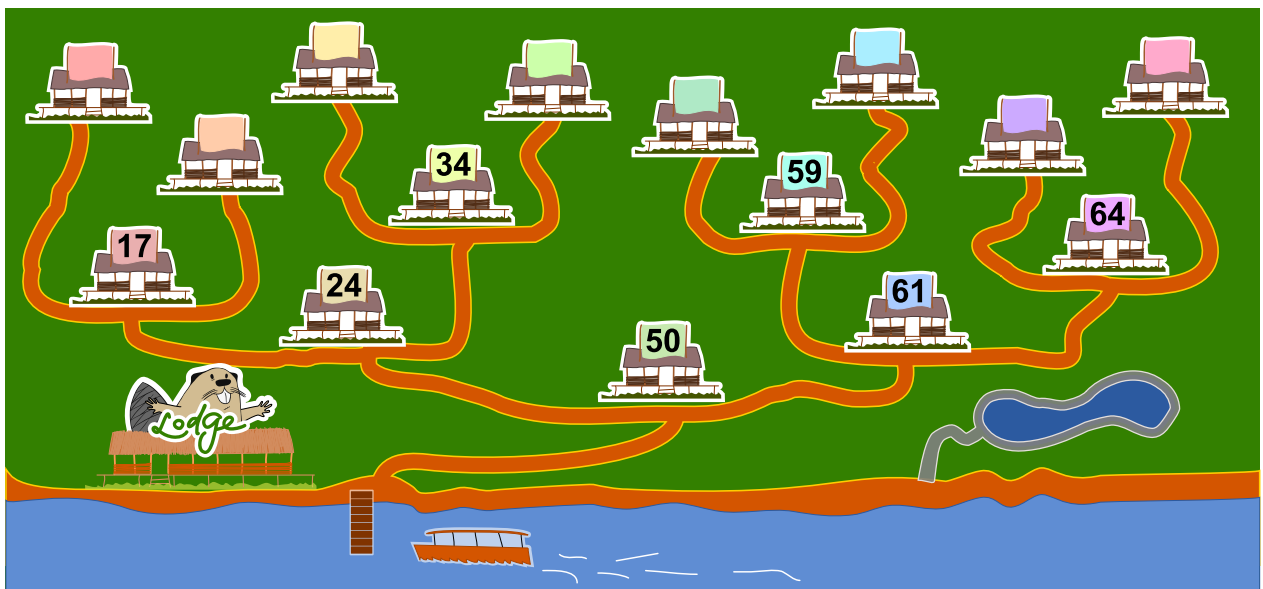
- [https://de.wikipedia.org/wiki/Division\\_mit\\_Rest#Modulo](https://de.wikipedia.org/wiki/Division_mit_Rest#Modulo)
- [https://de.wikipedia.org/wiki/Schriftliche\\_Division](https://de.wikipedia.org/wiki/Schriftliche_Division)
- [https://en.wikipedia.org/wiki/Euclidean\\_division](https://en.wikipedia.org/wiki/Euclidean_division)
- <https://de.wikipedia.org/wiki/RSA-Kryptosystem>



## 11. Ferienhaus Nr. 29

Milo macht ein Praktikum in einer Ferienhaus-Siedlung. Heute soll er Nummern an den Ferienhäusern anbringen. Einige Häuser sind bereits beschriftet. Er startet bei Haus 50. Von dort aus soll er...

- ...nach links gehen, wenn die neue Nummer kleiner ist als die Nummer des Hauses, bei dem er steht, ...
- ...nach rechts gehen, wenn die neue Nummer grösser ist als die Nummer des Hauses, bei dem er steht, ...
- ...die neue Ferienhaus-Nummer anbringen, wenn das Haus unbeschriftet ist.

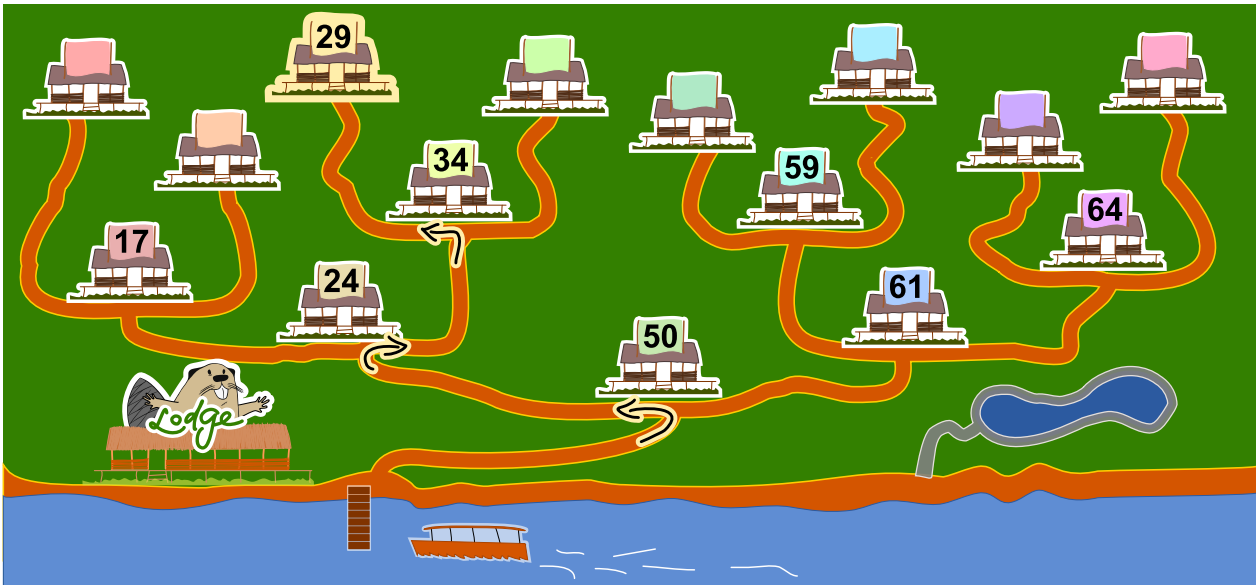


*An welchem Haus muss Milo die neue Nummer 29 befestigen?*



## Lösung

Das richtige Ferienhaus ist das dritte von links:



Die neue Ferienhausnummer 29 ist kleiner als die Nummer von Ferienhaus 50, also muss er als erstes nach links gehen. Im Vergleich zur Nummer von Ferienhaus 24 hingegen ist 29 grösser, also muss er nach rechts gehen. 29 ist wiederum kleiner als die Nummer von Ferienhaus 34, also muss er noch einmal nach links gehen. Das nächste Ferienhaus hat keine Nummer, also befestigt er dort die Nummer 29.

## Dies ist Informatik!

Die Nummerierung der Ferienhäuser entspricht einem *binären Suchbaum*, einer Datenstruktur die in der Informatik häufig genutzt wird. Mit einem binären Suchbaum kann man gespeicherte Daten schnell wiederfinden.

Ein binärer Suchbaum ist so aufgebaut, dass an jeder Kreuzung („Knoten“) ein „Element“ gespeichert ist. Nach jeder Kreuzung führen maximal zwei Wege („Kanten“) zu weiteren Kreuzungen. Beim Speichern von neuen Elementen wird z. B. stets der linke Weg eingeschlagen, wenn das neue Element einen kleineren Wert hat als das Element an der Kreuzung, sonst der rechte Weg. Das neue Element wird an der ersten freien Kreuzung gespeichert.

Beim Suchen nach einem Element kann man dann an jeder Kreuzung leicht entscheiden, welchen Weg man einschlagen muss. Wenn der binäre Suchbaum „balanciert“ ist (ein sogenannter *AVL-Baum*), muss man nach einem Schritt nur noch jeweils ungefähr die Hälfte der Kreuzungen durchsuchen. Das führt dazu, dass 1000 Elemente in nur 10 Schritten durchsucht werden können, 1'000'000 Elemente in 20 Schritten oder 1'000'000'000 Elemente in 30 Schritten (also  $n$  Elemente in  $\log_2(n)$  Schritten).

## Stichwörter und Webseiten

Binärer Suchbaum, AVL-Baum

- [https://de.wikipedia.org/wiki/Binärer\\_Suchbaum](https://de.wikipedia.org/wiki/Binärer_Suchbaum)
- <https://de.wikipedia.org/wiki/AVL-Baum>

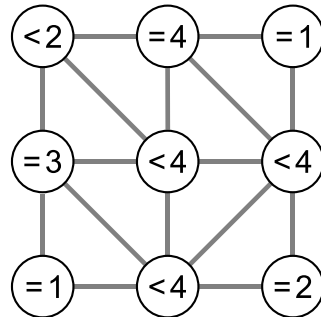




## 12. Nachbarn

Im Bild unten sind neun Kreise zu sehen, die teilweise miteinander verbunden sind. Eine Verbindung macht sie zu Nachbarn. Durch Anklicken können Kreise ausgewählt werden. Ein ausgewählter Kreis ist grün dargestellt, ein nicht ausgewählter Kreis weiss.

In jedem Kreis steht ein Ausdruck, der anzeigt, wie viele seiner Nachbarn ausgewählt werden sollen. „= 3“ bedeutet, dass genau drei der Nachbarn ausgewählt sein sollen. „< 4“ bedeutet, dass maximal drei der Nachbarn ausgewählt sein sollen.

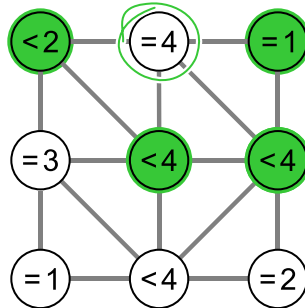


Wähle die Kreise so aus, dass die Bedingungen in allen neun Kreisen gleichzeitig erfüllt sind.

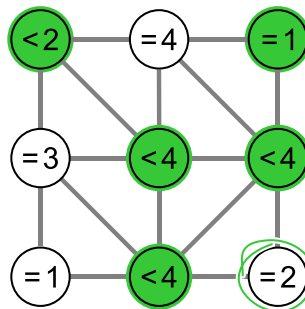


## Lösung

Der Kreis oben in der Mitte hat ein „= 4“ und somit genau vier Nachbarn, also müssen alle vier ausgewählt sein:



Ebenso hat der Kreis unten rechts ein „= 2“ und genau zwei Nachbarn, also müssen beide ausgewählt sein (der Kreis in der Mitte rechts ist bereits ausgewählt):



Damit sind bereits alle Bedingungen in allen neun Kreisen erfüllt.

Wenn man einen weiteren Kreis auswählen würde, würden Bedingungen verletzt:

- Wenn der „= 4“-Kreis oben in der Mitte ausgewählt würde, wäre die Bedingung des „= 1“-Kreises oben rechts nicht mehr erfüllt.
- Wenn der „= 3“-Kreis in der Mitte links ausgewählt würde, wäre die Bedingung des „< 2“-Kreises oben links nicht mehr erfüllt.
- Wenn der „= 1“-Kreis unten links ausgewählt würde, wäre die Bedingung des „= 3“-Kreises in der Mitte links nicht mehr erfüllt.
- Wenn der „= 2“-Kreis unten rechts ausgewählt würde, wäre die Bedingung des „< 4“-Kreises in der Mitte rechts nicht mehr erfüllt.

## Dies ist Informatik!

Wie viele Versuche braucht man, um das Problem zu lösen? Wenn man einfach alle möglichen Lösungen ausprobiert, hat man für jeden der 9 Kreise unabhängig 2 verschiedene Einstellungen, also  $2^9 = 512$  verschiedene Möglichkeiten. Einfach alle Möglichkeiten auszuprobieren, nennt man einen *brute-force-Ansatz*. Für jede dieser Möglichkeiten müsste dann überprüft werden, ob die Bedingungen erfüllt sind.

Sinnvoller ist es in diesem Fall jedoch, logisch und konsequent vorzugehen. Man sucht zuerst Kreise, deren Bedingungen eindeutig erfüllt werden können. Das sind zum Beispiel alle Kreise, die die



Bedingung „ $= n$ “ haben und genau  $n$  Verbindungen, also  $n$  Nachbarn haben. Von da an könnte man mit logischem Folgern weitermachen: schauen, ob es unerfüllte Bedingungen gibt, die man nur auf eine Art und Weise erfüllen kann. So kann man mit viel weniger Aufwand zur richtigen Lösung kommen. Im allgemeinen Fall kann man so auch feststellen, dass es keine Lösung gibt oder zumindest eine von mehreren Möglichkeiten findet. Eine analytische Vorgehensweise, bei der von allen Lösungsmöglichkeiten nur die vielversprechenden betrachtet werden, wird als *heuristisch* bezeichnet.

## Stichwörter und Webseiten

Nachbarschaft in Graphen, logisches Vorgehen

- [https://de.wikipedia.org/wiki/Nachbarschaft\\_\(Graphentheorie\)](https://de.wikipedia.org/wiki/Nachbarschaft_(Graphentheorie))
- <https://de.wikipedia.org/wiki/Heuristik#Informatik>
- <https://de.wikipedia.org/wiki/Brute-Force-Methode>





## 13. Computerspiel

Andrea hat ein Computerspiel in der Schule programmiert. Die Spielregeln sind ganz einfach:

Das Spiel besteht aus mehreren Spielrunden. In jeder Spielrunde fällt ein Blatt. Der Biber versucht das Blatt zu fangen, bevor es den Boden erreicht. Um zu gewinnen, muss der Biber 15 Blätter fangen, bevor 4 Blätter den Boden berühren.

Die Länge des Spiels wird in der Anzahl der Spielrunden gemessen.

Im folgenden Beispiel verliert der Biber nach 6 Spielrunden, weil das Maximum von 4 nicht gefangenen Blättern erreicht ist. Die Länge dieses Beispiels beträgt 6 Spielrunden.



Spielrunde	Resultat	Spielstand – Total Anzahl Blätter	
		Gefangen	Nicht gefangen
1	gefangen	1	0
2	nicht gefangen	1	1
3	gefangen	2	1
4	nicht gefangen	2	2
5	nicht gefangen	2	3
6	nicht gefangen	2	4

*Wie lange kann ein Spiel maximal dauern?*

- A) 4 Spielrunden
- B) 15 Spielrunden
- C) 18 Spielrunden
- D) 19 Spielrunden
- E) 20 Spielrunden
- F) Die Spiellänge ist unbegrenzt.



## Lösung

Um das längstmögliche Spiel zu finden, müssen wir alle Situationen kombinieren, in denen das Spiel weitergeht. Dazu kombinieren wir die maximal gefangenen Blätter vor Spielende (14 Spielrunden) mit den maximal nicht gefangenen Blättern vor Spielende (3 Spielrunden). Danach wird entweder ein 15. Blatt gefangen oder ein 4. Blatt verloren. Daher ist die maximale Länge  $15 + 3 = 14 + 4 = 18$  Runden und die richtige Antwort ist C).

Die Antwort A) „4 Runden“ wäre die Mindestlänge des Spiels (wenn alle Blätter nicht gefangen werden).

Die Antwort B) wäre die Mindestlänge, um das Spiel zu gewinnen (wenn alle Blätter gefangen werden).

Die Antworten D), E) und F) sind falsch, da das Maximum der gefangenen oder das Maximum der nicht gefangenen Blätter vorher erreicht würde.

## Dies ist Informatik!

Bei der Programmierung eines Spiels müssen die Regeln klar definiert sein. Die Auswirkungen der Regeln müssen vollständig verstanden werden, so dass das Spiel so aufgebaut werden kann, dass Gewinnen oder Verlieren möglich ist (ausreichende Blätter sind verfügbar), und dass das Spiel weder zu kurz noch zu lang dauert.

Ein Spiel, das aus mehreren Runden besteht, ist ein Prozess. Informatiker sind Spezialisten in der Modellierung und Beschreibung von Prozessen. Eine der Hauptaufgaben besteht darin herauszufinden, was alles passieren kann, und wie lange ein Prozess laufen kann.

## Stichwörter und Webseiten

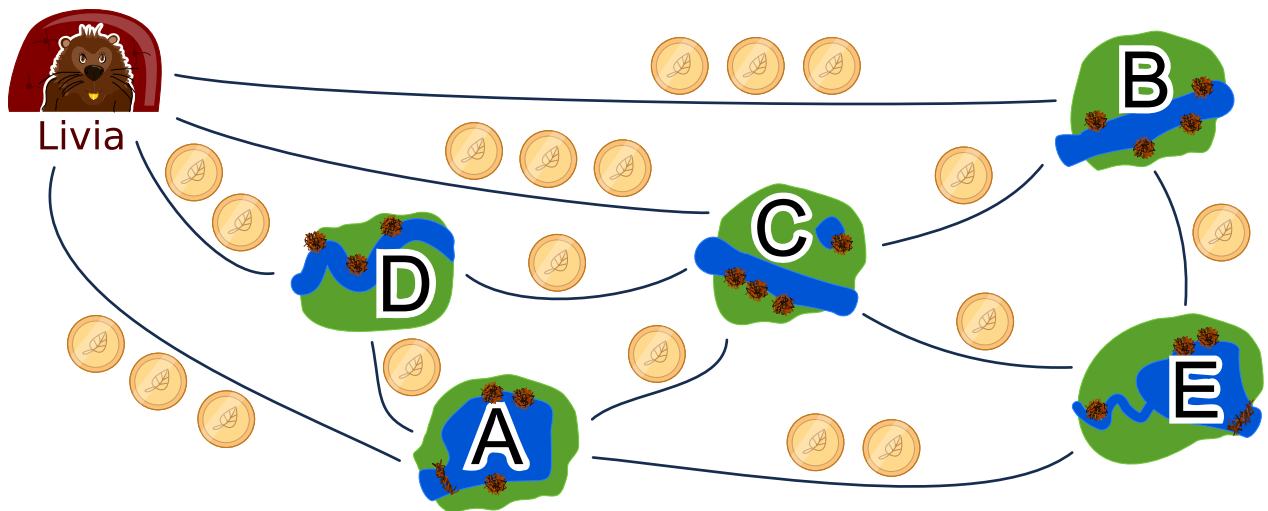
Analyse, Verifizierung und Validierung von Software

- [https://en.wikipedia.org/wiki/Software\\_verification](https://en.wikipedia.org/wiki/Software_verification)
- [https://de.wikipedia.org/wiki/Verifizierung\\_und\\_Validierung](https://de.wikipedia.org/wiki/Verifizierung_und_Validierung)



## 14. Biberbesuch

Livia möchte alle ihre Freunde in den Dörfern A, B, C, D und E mit öffentlichen Verkehrsmitteln besuchen. Sie besucht alle ihre Freunde auf einer einzigen Reise, ohne ein Dorf mehr als einmal zu besuchen. Am Ende ihrer Reise kehrt sie nach Hause zurück. Der Fahrpreis jeder Linie ist unten angezeigt.



Ein möglicher Weg, ihre Freunde zu besuchen ist:

Start → B → E → A → D → C → Start.

Dieser Weg kostet  $3 + 1 + 2 + 1 + 1 + 3 = 11$  Biber Münzen.

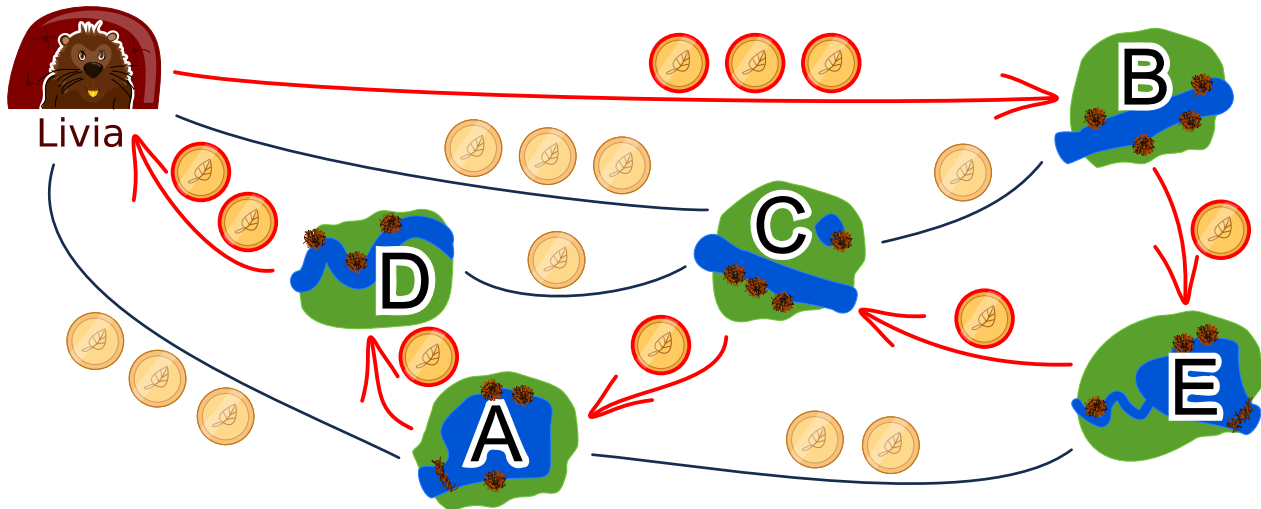
*In welcher Reihenfolge muss Livia die Freunde besuchen, damit sie möglichst wenige Münzen bezahlen muss?*



## Lösung

Es gibt zwei optimale Lösungen:

- Start → B → E → C → A → D → Start
- Start → D → A → C → E → B → Start



Die zwei Lösungen sind bis auf die Richtung gleich und kosten 9 Biber Münzen. Es gibt keine bessere Lösung, denn von Livias Zuhause aus kannst du einmal den Zwei-Biber Münzen-Weg und dann einen Drei-Biber Münzen-Weg gehen. Zu den vier weiteren Knoten gehören vier Wege, die jeweils mindestens eine Biber Münze kosten, was bereits 9 Biber Münzen ergibt.

Alle anderen Lösungen kosten mehr:

- Kosten von 10 Biber Münzen: Start → A → D → C → E → B → Start
- Kosten von 10 Biber Münzen: Start → A → E → B → C → D → Start
- Kosten von 10 Biber Münzen: Start → B → C → E → A → D → Start
- Kosten von 10 Biber Münzen: Start → B → E → A → C → D → Start
- Kosten von 10 Biber Münzen: Start → B → E → C → D → A → Start
- Kosten von 10 Biber Münzen: Start → C → B → E → A → D → Start
- Kosten von 10 Biber Münzen: Start → D → A → E → B → C → Start
- Kosten von 10 Biber Münzen: Start → D → A → E → C → B → Start
- Kosten von 10 Biber Münzen: Start → D → C → A → E → B → Start
- Kosten von 10 Biber Münzen: Start → D → C → B → E → A → Start
- Kosten von 11 Biber Münzen: Start → B → E → A → D → C → Start
- Kosten von 11 Biber Münzen: Start → C → D → A → E → B → Start

Eine Methode, den günstigsten Rundweg zu finden, besteht darin, einen Weg zu gehen, der die minimale Menge an Biber Münzen kostet, und dann von dort aus eine Lösung zu finden.





## Dies ist Informatik!

Nach guten oder sogar optimalen Lösungen zu suchen, ist eine der grundlegenden Aufgaben der Informatik. Wir können die Beschreibung dieser Optimierungsaufgabe in einem Graphen visualisieren, in dem Freunde Knoten und die Strassen Kanten sind. Die Aufgabe besteht darin, alle Knoten genau einmal so zu besuchen, dass die Summe der Kantengewichte (die Kosten in Bibermünzen) minimal sind. Dies ist ähnlich dem berühmten Travelling Salesman Problem (TSP).

Diese Art von Problemen sind normalerweise sehr schwierig mit einem Computer zu lösen. Um zu vermeiden, dass jede einzelne Lösung ausprobiert werden muss, kann man eine gute Heuristik verwenden (eine Heuristik ist zum Beispiel, zuerst den kürzesten Weg zu nehmen) und alle Lösungen, die schlechter werden, zu streichen. In diesem Fall erlauben wir, dass jeder Knoten nur einmal besucht werden darf. Wenn wir einen Knoten mehr als einmal besuchen dürfen, wird das Problem tatsächlich schwieriger, weil wir viel mehr Alternativen in Betracht ziehen müssen.

## Stichwörter und Webseiten

Optimierung, Problem eines Handlungsreisenden

- [https://de.wikipedia.org/wiki/Problem\\_des\\_Handlungsreisenden](https://de.wikipedia.org/wiki/Problem_des_Handlungsreisenden)
- <https://de.wikipedia.org/wiki/Optimierungsproblem>

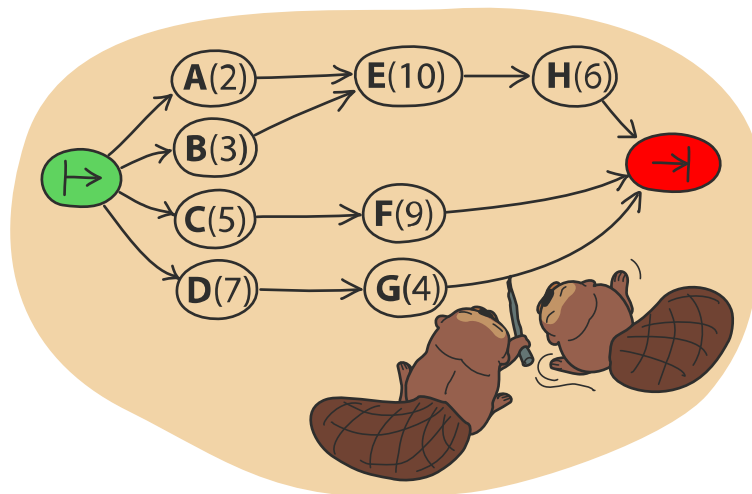




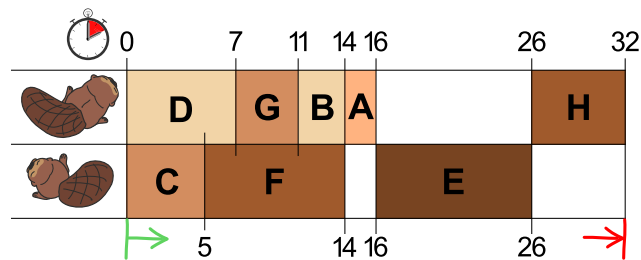
## 15. Zwei Biber bei der Arbeit

Zwei Biber bauen einen Damm und müssen dazu acht Aufgaben lösen: Bäume fällen, von den Stämmen die Äste entfernen, Stämme ins Wasser bringen, und so weiter. Für jede Aufgabe gibt es einen Buchstaben als Namen und eine Zahl in Klammern, die die nötige Anzahl der Arbeitsstunden angibt.

Einige Aufgaben können erst dann begonnen werden, wenn bestimmte andere Aufgaben bereits vollständig gelöst worden sind. Diese Abfolge wird durch die Pfeile dargestellt. Die Biber können parallel verschiedene Aufgaben bearbeiten, es kann aber immer nur einer an einer Aufgabe arbeiten.



Die Abbildung unten zeigt einen möglichen Arbeitsplan der beiden Biber, der 32 Stunden benötigt. Es geht aber schneller!



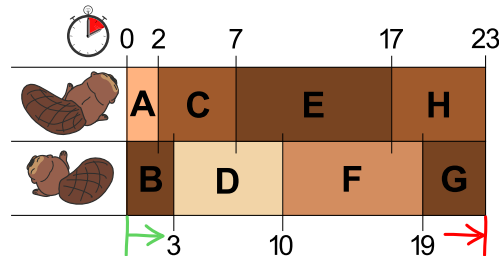
Was ist die kürzeste Zeit, in der die Biber einen Damm bauen können?



## Lösung

Es sind mindestens 23 Stunden erforderlich.

Das Bild in der Aufgabe zeigt einen möglichen Arbeitsplan der beiden Biber. Dort hat der erste Biber eine lange Pause von 10 Stunden und der zweite Biber zwei Pausen über insgesamt 8 Stunden. Wenn beide die ganze Zeit arbeiten würden, wären sie schneller fertig.



Wenn man darauf achtet, dass die beiden grössten Aufgaben E(10) und F(9) nicht von demselben Biber ausgeführt werden, findet man leicht einen Arbeitsplan, der mit 23 Stunden auskommt. Schneller geht es nicht, denn die beiden Biber arbeiten ohne Pause.

## Dies ist Informatik!

Um einen kürzesten Arbeitsplan zu finden, wäre eine Möglichkeit, sich an die folgende Regel zu halten: „Wähle unter den noch verfügbaren Aufgaben immer die mit den meisten Arbeitsstunden“. In der Informatik nennt man eine solche Strategie „greedy“ (engl. für „gierig“). Man löst zuerst die Teilaufgaben, die einen möglichst grossen Fortschritt im Hinblick auf die Gesamtlösung des Problems bedeuten.

In vielen Fällen ist „greedy“ eine gute Strategie, aber manchmal – wie bei dieser Aufgabe – funktioniert sie nicht so gut. Diese Aufgabe wurde absichtlich so konstruiert, dass die „greedy“-Strategie nicht funktioniert. Das Finden solcher ungünstigen Problemstellungen ist jedoch auch wichtig: In der theoretischen Informatik beispielsweise sucht man für Computerprogramme gezielt nach dem ungünstigsten Fall („worst case“), um den Zeitbedarf von Algorithmen besser abschätzen zu können. Eigentlich gäbe es nur einen sicheren Weg, die beste Lösung zu finden: Man probiert alle denkbaren Arbeitspläne aus, die den vorgegebenen Regeln entsprechen. Bei grösseren Projekten kann aber die Anzahl der Möglichkeiten so gross sein, dass die Entscheidung zu viel Zeit benötigt. Da kommt dann eine Strategie wie „greedy“ ins Spiel, denn mit ihr kann man einfach Lösungen finden, die zumindest hinreichend gut sind.

## Stichwörter und Webseiten

Scheduling, Greedy-Algorithmus

- <https://de.wikipedia.org/wiki/Scheduling>
- [https://de.wikipedia.org/wiki/Topologische\\_Sortierung](https://de.wikipedia.org/wiki/Topologische_Sortierung)
- <https://de.wikipedia.org/wiki/Greedy-Algorithmus>



## A. Aufgabenautoren

 Andrea Adamoli	 Wei-fu Hou	 Ilya Posov
 Jared Asuncion	 Juraj Hromkovič	 Nol Premasathian
 Javier Bilbao	 Takeharu Ishizuka	 J.P. Pretti
 Lucia Budinská	 Svetlana Jakšič	 Doris Reck
 Špela Cerar	 Dong Yoon Kim	 Kirsten Schlüter
 Kris Coolsaet	 Vaidotas Kinčius	 Andrea Maria Schmid
 Valentina Dagienė	 Jia-Ling Koh	 Mohamed El-Sherif
 Darija Dasović Rakijašić	 Regula Lacher	 Jacqueline Staub
 Christian Datzko	 Dan Lessner	 Jacqueline Staub
 Susanne Datzko	 Dimitris Mavrovouniotis	 Allira Storey
 Marissa Engels	 Karolína Mayerová	 Peter Tomcsányi
 Hanspeter Erni	 Samart Moodleah	 Willem van der Vegt
 Georgios Fessakis	 Tom Naughton	 Jiří Vaníček
 Gerald Futschek	 Sanja Pavlovic Šijanović	 Troy Vasiga
 Martin Guggisberg	 Péter Piltmann	 Michael Weigend
 Bent Halden	 Zsuzsa Pluhár	 Magdalena Zarach
 Urs Hauser	 Wolfgang Pohl	



## B. Sponsoring: Wettbewerb 2018

### 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.zubler.ch/>

Zubler & Partner AG Informatik

Umfassendes Angebot an Dienstleistungen.



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

OXOcard: Spielend programmieren lernen

OXON



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

Diartis AG

Diartis entwickelt und vertreibt Softwarelösungen für das Fallmanagement.



<http://senarclens.com/>

Senarclens Leu & Partner



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

Ausbildungs- und Beratungszentrum für Informatikunterricht der ETH Zürich.



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

Haute école pédagogique du canton de Vaud



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

Pädagogische Hochschule Luzern



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

Pädagogische Hochschule FHNW



<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](http://www.svia-ssie-ssii.ch)  
schweizerischer vereinfürinformatikind  
erausbildung//sociétésuissepourl'infor  
matique dans l'enseignement//societàsviz  
zera per l'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.