INFORMATIK-BIBER SCHWEIZ
CASTOR INFORMATIQUE SUISSE
CASTORO INFORMATICO SVIZZERA

Aufgaben und Lösungen 2020

Schuljahre 3/4

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

Herausgeber:

Susanne Datzko, Fabian Frei, Juraj Hromkovič,
Regula Lacher, Jean-Philippe Pellet

0101001101010110010010010000001001011010101001101010010100100101000010

SV!A

www.svia-ssie-ssii.ch
schweizerischerverein für informatikinderausbildung//société suisse pour l'informatique dans l'enseignement//società svizzera per l'informatica nell'insegnamento

Mitarbeit Informatik-Biber 2020

Susanne Datzko, Fabian Frei, Martin Guggisberg, Lucio Negrini, Gabriel Parriaux, Jean-Philippe Pellet

Projektleitung: Nora A. Escherle

Herzlichen Dank für die Aufgabenentwicklung für den Schweizer-Wettbewerb an:
Juraj Hromkovič, Michael Barot, Christian Datzko, Jens Gallenbacher, Dennis Komm, Regula Lacher, Peter Rossmanith: ETH Zürich, Ausbildunges- und Beratungszentrum für Informatikunterricht

Die Aufgabenauswahl wurde erstellt in Zusammenarbeit mit den Organisatoren von Bebras in Deutschland, Österreich, Ungarn, Slowakei und Litauen. Besonders danken wir:
Valentina Dagienė: Bebras.org
Wolfgang Pohl, Hannes Endreß, Ulrich Kiesmüller, Kirsten Schlüter, Michael Weigend: Bundesweite Informatikwettbewerbe (BWINF), Deutschland
Wilfried Baumann, Anoki Eischer: Österreichische Computer Gesellschaft
Gerald Futschek, Florentina Voboril: Technische Universität Wien
Zsuzsa Pluhár: ELTE Informatikai Kar, Ungarn
Michal Winzcer: Comenius University, Slowakei

Die Online-Version des Wettbewerbs wurde auf cuttle.org realisiert. Für die gute Zusammenarbeit danken wir:
Eljakim Schrijvers, Justina Dauksaite, Arne Heijenga, Dave Oostendorp, Andrea Schrijvers, Alieke Stijf, Kyra Willekes: cuttle.org, Niederlande
Chris Roffey: University of Oxford, Vereinigtes Königreich

Für den Support während den Wettbewerbswochen danken wir:
Hanspeter Erni: Schulleitung Sekundarschule Rickenbach
Gabriel Thullen: Collège des Colombières
Beat Trachsler: Kantonsschule Kreuzlingen

Christoph Frei: Chragokyberneticks (Logo Informatik-Biber Schweiz)
Dr. Andrea Leu, Maggie Winter, Brigitte Manz-Brunner: Senarclens Leu + Partner AG

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

Die französischsprachige Übersetzung wurde von Elsa Pellet und die italienischsprachige Übersetzung von Christian Giang erstellt.

INFORMATIK-BIBER SCHWEIZ
CASTOR INFORMATIQUE SUISSE
CASTORO INFORMATICO SVIZZERA

Der Informatik-Biber 2020 wurde vom Schweizerischen Verein für Informatik in der Ausbildung SVIA durchgeführt und von der Hasler Stiftung unterstützt.

HASLERSTIFTUNG

Dieses Aufgabenheft wurde am 9. September 2021 mit dem Textsatzsystem LaTeX erstellt. Wir bedanken uns bei Christian Datzko für die Entwicklung und langjährige Pflege des Systems zum Generieren der 36 Versionen dieser Broschüre (nach Sprachen und Schulstufen). Das System wurde analog zum Vorgänger-System neu programmiert, welches ab 2014 gemeinsam mit Ivo Blöchliger entwickelt wurde. Jean-Philippe Pellet danken wir für die Entwicklung der `bebras` Toolchain, die seit 2020 für die automatisierte Konvertierung der Markdown- und YAML-Quelldokumente verwendet wird.

Hinweis: Alle Links wurden am 1. Dezember 2020 geprüft.

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.bebras.org/`), die in Litauen ins Leben gerufen wurde.

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

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

Der Informatik-Biber 2020 wurde in fünf Altersgruppen durchgeführt:

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

In den Altersklassen 3 und 4 hatten 9 Aufgaben zu lösen, nämlich aus den drei Schwierigkeitsstufen leicht, mittel und schwer jeweils drei. Für die Altersklassen 5 und 6 waren es je vier Aufgaben aus jeder Schwierigkeitsstufe, also 12 insgesamt. Für die restlichen Altersklassen waren es 15 Aufgaben, nämlich fünf Aufgaben pro Schwierigkeitsstufe.

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

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

Dieses international angewandte System zur Punkteverteilung soll den Anreiz zum blossen Erraten der Lösung eliminieren.

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

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

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

Für weitere Informationen:

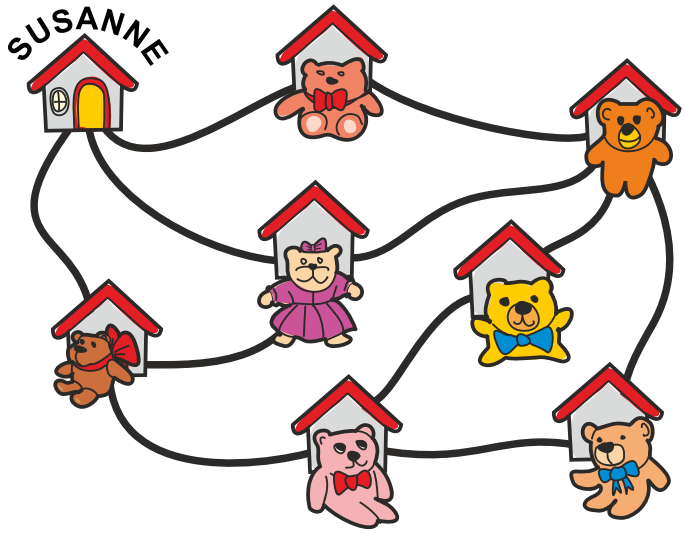
SVIA-SSIE-SSII Schweizerischer Verein für Informatik in der Ausbildung
Informatik-Biber
Nora A. Escherle

`https://www.informatik-biber.ch/de/kontaktieren/`
`https://www.informatik-biber.ch/`

Inhaltsverzeichnis

1. Teddybärenjagd

In Susannes Quartier sind folgende Teddybären vor den Häusern zu finden.



Susanne hat von ihrem eigenen Haus aus einen Rundgang an genau vier anderen Häusern vorbei gemacht. Sie ist an keinem Haus zweimal vorbeigegangen. Bei einem Haus hat sie den Teddybär übersehen. Die drei anderen Teddybären waren:

Welchen Teddybären hat Susanne übersehen?

A) 

B) 

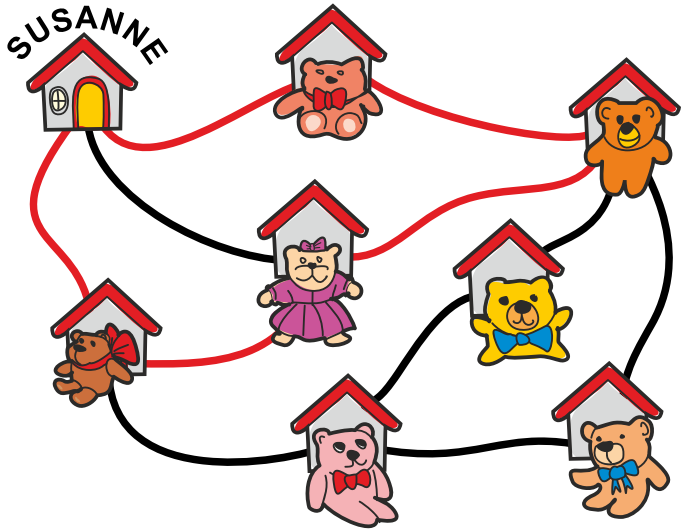
C) 

D) 

Lösung

Die richtige Antwort ist C) .

Bei ihrem Rundgang muss Susanne an den Häusern mit den drei Teddybären , und vorbeigegangen sein. Diese drei Teddybären sind direkt durch einen Weg verbunden. Zum ersten Teddybär kommt sie direkt von ihrem Daheim. Am Ende dieses Weges ist sie beim dritten Teddybär . Von dort aus gibt es nur einen Weg zurück zu ihrem Daheim, der über ein einziges viertes Haus geht, nämlich der Weg über den Teddybär . Andere mögliche Wege gehen an mindestens zwei weiteren Teddybären vorbei. Sie ist aber nur an vier Häusern vorbeigegangen. Die folgende Karte zeigt den Weg:



Susanne kann den Rundgang in beide möglichen Richtungen zurücklegen, das spielt keine Rolle.

Dies ist Informatik!

Lückentexte im Deutschunterricht, Mathematikaufgaben mit leeren Feldern oder eine Teddybärenjagd mit einem fehlenden Bild: das sind alles Aufgaben, bei denen eine fehlende Information gesucht ist. Die Aufgaben sind aber so aufgebaut (oder auch *strukturiert*), dass diese fehlende Information durch *logisches Denken* oder *Schlussfolgern* gefunden werden kann.

In der Informatik kommt so etwas immer wieder vor. Bei Datenübertragungen oder beim Speichern von Daten können Fehler passieren. Deshalb arbeitet man mit Verfahren zum *Erkennen von Fehlern* oder sogar zum *Korrigieren von Fehlern*. Wenn der Fehler nicht zu gross ist, schafft man das, indem man bewusst mehr Information speichert als eigentlich nötig. In dieser Aufgabe ist das die Karte und die Tatsache, dass Susanne genau an vier Teddybären vorbeiging. So kann sie die fehlende Information finden, nämlich welchen Teddybär sie übersehen hat.

Übrigens wurde in einigen Ländern der Welt im Jahr 2020 tatsächlich solche «Teddybärenjagden» (Englisch «Teddy Bear Hunt») angeboten, bei der in verschiedenen Häusern Teddybären versteckt wurde, die man von aussen her suchen konnte. Dies ermöglichte es den Kindern, sich gemeinsam am

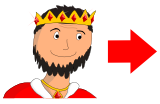
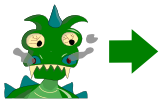
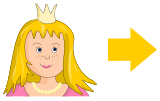
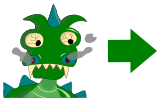
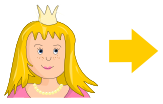
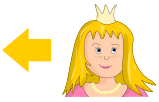
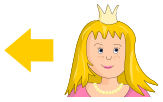
Verstecken und Entdecken zu erfreuen, trotz den Distanzregeln wegen des Corona-Virus. Die Idee, eine Bärenjagd nachzuspielen, stammt ursprünglich aus dem Bilderbuch «We're Going on a Bear Hunt» von Michael Rosen (1989). Bei uns kennt man das Buch und das zugehörige Spiel auch als «Wir gehen auf die Löwenjagd».

Stichwörter und Webseiten

- Fehlererkennung, Fehlerbehebung: `https://de.wikipedia.org/wiki/Fehlererkennung`
- Logisches Schlussfolgern
- Teddybärenjagd: `https://www.insider.com/coronavirus-pandemic-sparked-worldwide-bear-hunt-to-entertain-kids-2020-4`, `https://www.youtube.com/watch?v=0gyI6ykDwds`

2. Das Theaterstück

In einem Theaterstück spielen eine schöne Prinzessin, ein edler Ritter, der weise König und ein böser Drache mit. Am Anfang ist die Bühne leer. Während der Aufführung des Theaterstücks betreten und verlassen diese vier Figuren die Bühne in der folgenden Reihenfolge:

Erster Akt		**Zweiter Akt**
König betritt Bühne 		Drache betritt Bühne 
Prinzessin betritt Bühne 		Ritter betritt Bühne 
König verlässt Bühne 	P A U S E	Drache verlässt Bühne 
Drache betritt Bühne 		Prinzessin betritt Bühne 
Prinzessin verlässt Bühne 		Ritter verlässt Bühne 
Drache verlässt Bühne 		Prinzessin verlässt Bühne 
Pause		**Ende**

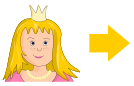
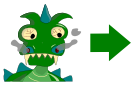
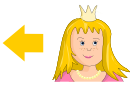
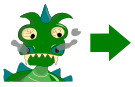
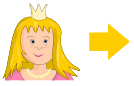
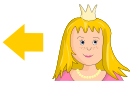
Was wird nicht passieren?

A) Die Prinzessin und der Ritter sind gemeinsam auf der Bühne.
B) Der König und der Drache sind gemeinsam auf der Bühne.
C) Der Ritter betritt die Bühne erst nach der Pause.
D) Der Ritter und der Drache sind gemeinsam auf der Bühne.

Lösung

Die richtige Antwort ist B) «Der König und der Drache sind gemeinsam auf der Bühne.», denn diese Behauptung stimmt während des ganzen Stückes nie.

Man kann sich das schrittweise überlegen:

Handlung	König auf Bühne?	Prinzessin auf Bühne?	Drache auf Bühne?	Ritter auf Bühne?	Übereinstimmung mit Behauptung der Antworten
Erster Akt					
	Ja	Nein	Nein	Nein	
	Ja	Ja	Nein	Nein	
	Nein	Ja	Nein	Nein	
	Nein	Ja	Ja	Nein	
	Nein	Nein	Ja	Nein	
	Nein	Nein	Nein	Nein	
Pause					
Zweiter Akt					
	Nein	Nein	Ja	Nein	
	Nein	Nein	Ja	Ja	C), D)
	Nein	Nein	Nein	Ja	
	Nein	Ja	Nein	Ja	A)
	Nein	Ja	Nein	Nein	
	Nein	Nein	Nein	Nein	
Ende					

Für jede Antwort kann geprüft werden, ob die darin gemachte Behauptung stimmt oder nicht, indem man die Zeilen der Tabellen durchgeht.

Bei der Antwort A) wird nach einer Zeile gesucht, in der sowohl die Prinzessin als auch der Ritter gemeinsam auf der Bühne sind. Das ist in der vierten Zeile des zweiten Aktes der Fall, denn dann betritt die Prinzessin die Bühne, wo der Ritter schon seit der zweiten Zeile ist und bis zur fünften Zeile bleibt. Die Behauptung von Antwort A) stimmt also zu mindestens einem Zeitpunkt.

Bei der Antwort D) wird nach einer Zeile gesucht, in der der Ritter und der Drache gemeinsam auf der Bühne sind. Das ist in der zweiten Zeile des zweiten Aktes der Fall, denn der Ritter betritt die Bühne in der zweiten Zeile, während der Drache die Bühne schon in der ersten Zeile betreten hat und bis zur dritten Zeile bleibt. Die Behauptung von Antwort D) stimmt also zu mindestens einem Zeitpunkt.

Bei der Antwort C) ist die Behauptung von einer anderen Art. Wenn diese stimmen soll, darf der Ritter die Bühne während des gesamten ersten Aktes nicht betreten haben. Hier muss man sich die Spalte des Ritters für den ersten Akt anschauen. Hier steht überall «Nein», also hat der Ritter die Bühne tatsächlich während des gesamten ersten Aktes nicht betreten. Er betritt sie dann aber in der zweiten Zeile des zweiten Aktes, also stimmt die Behauptung von Antwort C) ebenfalls.

Wenn die Behauptung von Antwort B) stimmen würde, müssten der König und der Drache in irgendeiner Zeile gemeinsam auf der Bühne stehen. Doch in keiner der zwölf Zeilen steht in beiden Spalten ein «Ja». Es ist sogar so, dass der König bereits in der dritten Zeile des ersten Aktes die Bühne verlässt und sie bis zum Ende nicht mehr betritt. Der Drache hingegen betritt die Bühne erst in der vierten Zeile des ersten Aktes. Vielleicht begegnen sich die beiden hinter der Bühne, aber auf der Bühne sind sie nie gemeinsam. Damit stimmt die Behauptung von Antwort B) nicht. Also ist B) die korrekte Antwort.

Dies ist Informatik!

Auch wenn man sich aufgrund des Ablaufs des Theaterstücks lebhaft eine Geschichte vorstellen kann, ist in dieser Aufgabe für jede Figur immer nur eine Eigenschaft wichtig: Befindet sie sich zu einem bestimmten Zeitpunkt auf der Bühne oder nicht? Dieses Einschränken des Blicks auf bestimmte Eigenschaften nennt man *Abstraktion*.

In der Informatik kann man solche Abstraktionen sehr gut formulieren. Für jede der vier Figuren definieren wir eine sogenannte *Variable*, die uns die Frage beantwortet, ob sich die Figur gerade auf der Bühne befindet. Die vier Variablen sind: «König auf Bühne?», «Prinzessin auf Bühne?», «Drache auf Bühne?» und «Ritter auf Bühne?». Während des Stückes ändern sich die Antworten auf diese Fragen immer wieder; für jede Frage ist die Antwort manchmal «ja» und manchmal «nein». In der Informatik nennen wir die aktuelle Antwort auf eine Frage den aktuellen *Wert* der zugehörigen Variablen. Der Wert einer Variablen kann sich in der Informatik also immer wieder ändern. (In der Mathematik ist das anders, dort ändern Variablen ihre Werte nicht über die Zeit hinweg.) Die Tabelle in der Antworterklärung zeigt die vier Variablen und die zugehörigen Werte zu jedem Zeitpunkt.

Es gibt noch eine andere Weise, das Theaterstück zu betrachten. Zu jedem Zeitpunkt schauen wir, welche Figuren gerade auf der Bühne sind. (Wir betrachten also die momentanen Werte der vier Variablen.) Jede mögliche Kombination von Figuren nennen wir einen *Zustand* der Bühne. Wenn

eine Figur die Bühne betritt oder verlässt, ändert sich also der Zustand der Bühne. Wir nennen das dann auch einen *Übergang* der Bühne von einem Zustand in einen anderen. Wenn man ein Blatt Papier nimmt und sich für jeden möglichen Zustand (also jede Figurenkombination) einen separaten Kreis zeichnet, kann man das Ganze als eine Abstraktion der Bühne betrachten.

Zusätzlich kann man noch die möglichen Übergänge als Pfeile einzeichnen, die von einem Zustand zu einem anderen führen. Wenn wir das auch noch tun, haben wir das, was wir in der Informatik den *Zustandsgraphen* der Bühne nennen würden.

Zu Beginn des Theaterstückes ist die Bühne leer. Den entsprechenden Zustand nennen wir deshalb *Anfangszustand*. Den Ablauf des Theaterstückes können wir jetzt als einen Weg im Zustandsgraphen einzeichnen. Der Weg beginnt im Anfangszustand und geht dann denjenigen Pfeilen entlang, die der Handlung entsprechen.

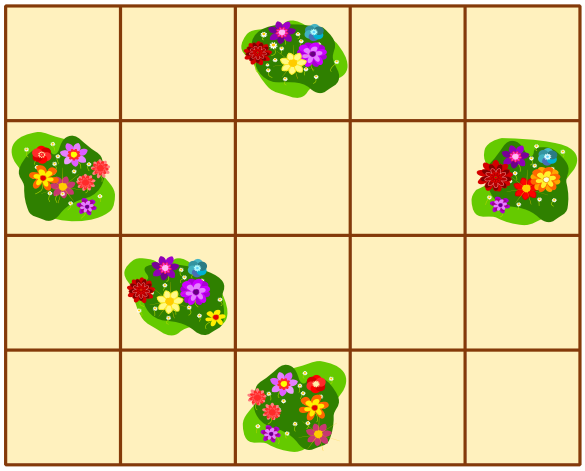
Zustandsgraphen sind in der Informatik sehr wichtig. Bei fast jedem komplizierten System muss man irgendwann über den Zustandsgraphen nachdenken. Für Menschen ist es aber oft sehr mühsam, mit solchen abstrakten Zuständen und Übergängen zu arbeiten. Computer können das hingegen extrem gut. Daher lohnt es sich, wenn Menschen ihre Probleme so mit Zustandsgraphen darstellen können, dass Computer sie dann lösen können.

Stichwörter und Webseiten

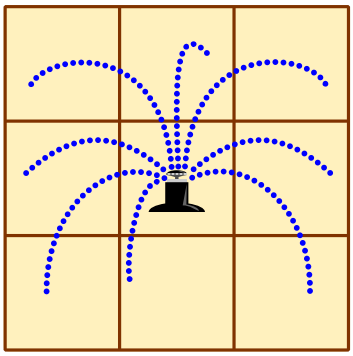
- Variablen: `https://de.wikipedia.org/wiki/Variable_(Programmierung)`
- Zustände, Übergänge, Zustandsgraph: `https://de.wikipedia.org/wiki/Zustandsübergangsdiagramm`

3. Beete bewässern

Daniels Garten besteht aus quadratischen Feldern. In einigen dieser Felder hat er Blumen gepflanzt.



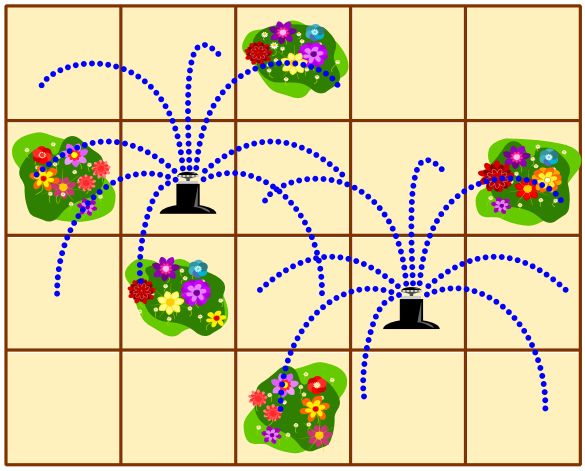
Im Sommer möchte er die Blumen mit Rasensprengern bewässern. Auf die Felder mit Blumen kann er keinen Rasensprenger stellen. Ein Rasensprenger bewässert alle Blumen in den 8 Feldern um ihn herum:



Platziere so wenige Rasensprenger wie nötig, um alle Blumenfelder zu bewässern.

Lösung

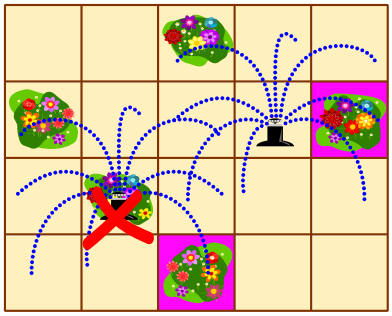
Die folgende Lösung braucht zwei Rasensprenger, um alle Blumenfelder zu bewässern:



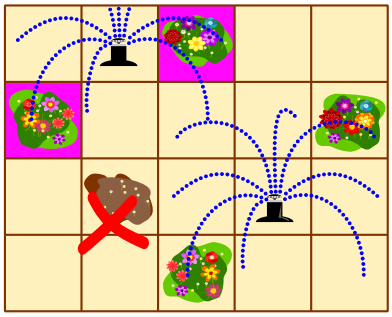
Zwischen dem Blumenfeld ganz links und dem Blumenfeld ganz rechts liegen drei Felder. Ein einzelner Rasensprenger kann keine Felder bewässern, die so weit auseinander liegen.

Es gibt auch keine weitere Lösung mit nur zwei Rasensprengern.

Um das Blumenfeld ganz rechts und das unten in der Mitte gleichzeitig zu bewässern, muss ein Rasensprenger genau da stehen, wo er in der Lösung steht. Wenn er eines höher stehen würde, um das Blumenfeld oben in der Mitte auch noch zu bewässern, würde er das Blumenfeld unten in der Mitte nicht mehr bewässern und man könnte die verbleibenden drei Blumenfelder nicht mit einem Rasensprenger bewässern, denn auf einem Blumenfeld darf kein Rasensprenger stehen.



Um das Blumenfeld ganz links und das oben in der Mitte zu bewässern, müsste ein Sprinkler entweder so stehen, wie in der Lösung gezeigt, oder ein Feld darüber. Wenn dieser Sprinkler aber zusätzlich noch das Blumenfeld in der zweiten Spalte von links und in der dritten Zeile von oben bewässern soll, kann er nicht ganz oben stehen.



Dies ist Informatik!

Diese Aufgabe ist ein typisches Optimierungsproblem: Während klar ist, dass alle Blumenfelder bewässert werden sollen, ist die Anzahl der benötigten Rasensprenger variabel und sollte möglichst klein sein. Ähnliche Optimierungsprobleme tauchen auf, wenn man beispielsweise Ortschaften mit Feuerwehrstationen absichern oder Höfe mit Natelempfang versorgen möchte.

In der Informatik spricht man auch von *Überdeckungsproblemen*. Diese gehören zu einer Klasse von sehr schwierigen Problem in der Informatik. Die richtige Platzierung einer minimalen Anzahl von Rasensprengern war in der Aufgabe zwar noch recht einfach. Doch die Schwierigkeit steigt mit der Anzahl an Blumenfeldern so stark an, dass man bald einmal keine optimale Lösung mehr finden kann in vernünftiger Zeit, selbst mit Computerunterstützung.

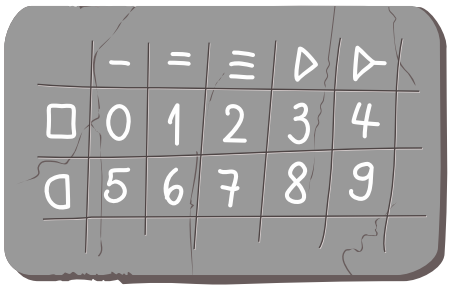
Eine Möglichkeit in solchen Fällen ist es dann, dass man sich mit Lösungen zufrieden gibt, die vielleicht nicht optimal sind, aber immer noch gut. Es macht keinen grossen Unterschied, ob man jetzt 101 statt nur 100 Feuerwehrstationen oder 1000 Natelsendemasten statt nur 990 positioniert; das Problem ist dadurch aber oft viel leichter zu lösen.

Stichwörter und Webseiten

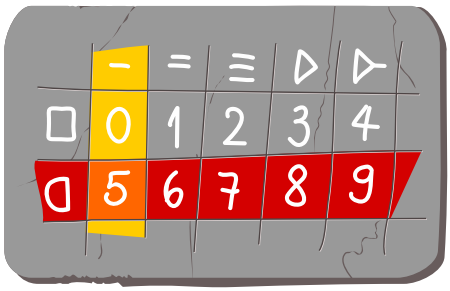
- Optimierung: `https://de.wikipedia.org/wiki/Optimierungsproblem`
- Überdeckungsproblem

4. Baujahr der Biberburg

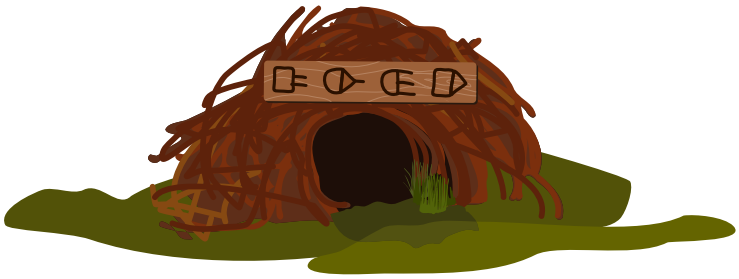
Auf dem Schild über dem Eingang jeder Biberburg steht das Baujahr. Die Biber verwenden für die Ziffern eigene Zeichen. Die Tabelle rechts zeigt, wie man aus den Ziffern die Zeichen der Biber zusammensetzen kann:



Beispielsweise setzen die Biber die Ziffer «5» so zu dem neuen Zeichen  zusammen:



So sieht Cleverias Biberburg aus:



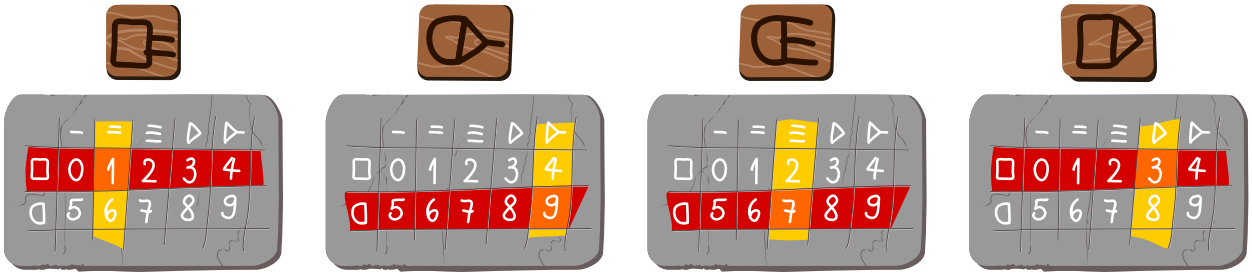
In welchem Jahr wurde Cleverias Biberburg gebaut?

A) 0978
B) 1574
C) 1923
D) 1973
E) 1993
F) 2973
G) 6378

Lösung

Das Baujahr der Biberburg findest du heraus, indem du für jedes Symbol die entsprechende Zeile und die entsprechende Spalte bestimmst. An der Kreuzung der Spalte und der Zeile findest du die gesuchte Ziffer.

Weil es vier Symbole sind, machst du das vier Mal.



Die vier Ziffern in der richtigen Reihenfolge ergeben die Zahl 1973.

Dies ist Informatik!

Das Geheimhalten von Informationen und das Schützen von Daten ist eine 4000 Jahre alte Aufgabe. Unzählige Geheimsprachen wurden zu diesem Zweck entwickelt und benutzt. Heute ist Datensicherheit eines der Kernthemen der Informatik. Eine der Methoden, Daten vor unbefugtem Lesen zu schützen, ist sie *zu chiffrieren*. Das Chiffrieren verwandelt einen *Klartext* in einen *Geheimtext*. Das Rekonstruieren des Klartextes aus dem Geheimtext nennt man *Dechiffrieren*. Die Lehre der Geheimschriften nennt man *Kryptologie*.

Die antiken Kulturen verwendeten meistens Geheimschriften, die durch Codierung von Buchstaben mit anderen Buchstaben oder ganz neuen Zeichen erzeugt worden sind. Die Geheimschrift hier ist speziell für den Informatik-Biber entwickelt worden, basiert aber auf einem Konzept aus dem antiken Palästina. Die damalige Sicherheitsregel war, dass nur Geheimschriften verwendet worden sind, die man leicht auswendig lernen kann. Eine schriftliche Beschreibung der Geheimschrift aufzubewahren, betrachtete man als zu grosses Risiko. Eine Tabelle, wie sie hier verwendet wird, kann man gut auswendig lernen. Die berühmte Geheimschrift der Freimaurer basiert auf diesem Prinzip.

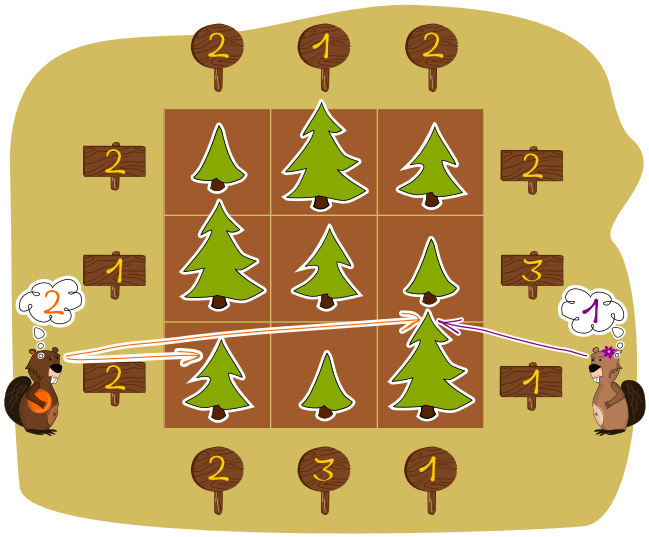
Statt nur neue Zeichen für Ziffern zusammenzustellen, kann man kann auch eigene neue Geheimschriften für Texte erfinden. Dazu schreibt man in eine Tabelle alle Buchstaben und erfindet neue Symbole für die Spalten und Zeilen und erhält so für alle Buchstaben neue Zeichen.

Stichwörter und Webseiten

- Kryptographie (Substitution): `https://de.wikipedia.org/wiki/Kryptographie`
- Geheimschriften

5. 3×3-Tannen-Sudoku

Biber pflanzen Tannen in Reihen. Die Tannen haben drei unterschiedliche Höhen (1 , 2 und 3) und in jeder Reihe gibt es genau eine Tanne von jeder Höhe.



Wenn sich die Biber eine Tannenreihe von einem Ende her anschauen, dann können sie niedrigere Tannen, die hinter höheren Tannen versteckt sind, **nicht** sehen.

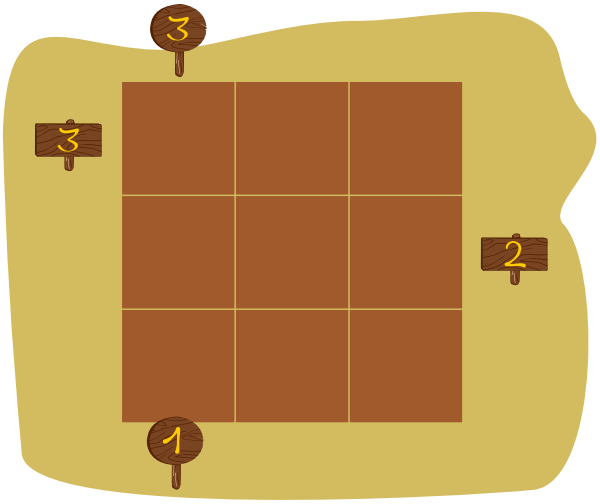
Am Ende jeder Tannenreihe steht auf einem Schild, wie viele Tannen ein Biber von dieser Stelle sehen kann.

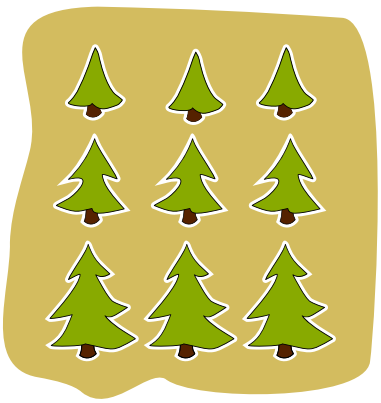
Nun pflanzen die Biber neun Tannen in ein 3×3-Feld, wie im Beispiel rechts.

Dabei gelten folgende Regeln:

- In jeder Zeile (horizontalen Reihe) gibt es genau eine Tanne von jeder Höhe.
- In jeder Spalte (vertikalen Reihe) gibt es genau eine Tanne von jeder Höhe.
- Die Schilder mit der Anzahl sichtbarer Tannen stehen rund um das 3×3-Feld.

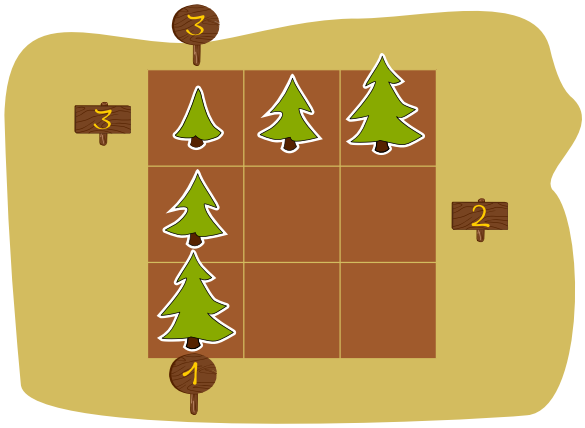
Verteile die Tannen auf die richtigen Felder.





Lösung

Im Feld zeigen zwei Schilder, dass von diesen Positionen drei Tannen gesehen werden können. Alle drei Tannen einer Reihe kann man nur sehen, wenn die Tannen so geordnet sind, dass ihre Höhe ansteigt, also von dieser Position weg. Damit sind die Spalte links und die oberste Zeile bestimmt:

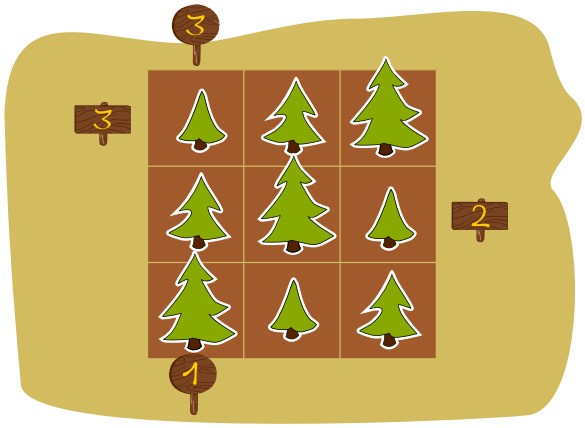


Das Schild rechts mit der 2 verlangt, dass von dort zwei Tannen sichtbar sind, also muss ganz in der Mitte eine Tanne der Höhe 3 sein und diese mittlere Zeile ist somit (), 3 (), 1 ().

Die weiteren Felder werden gemäss der «Sudoku»-Regel gefüllt, dass von jeder Höhe genau eine Tanne in jeder Reihe sein muss.

In der Mitte der untersten Zeile muss eine Tanne der Höhe 1 () stehen, weil für in der mittleren Spalte die beiden anderen Höhen bereits vergeben sind. Ganz rechts unten muss schliesslich eine Tanne der Höhe 2 () folgen, um die Reihe vollständig zu machen.

Die vollständige Lösung sieht so aus:



Dies ist Informatik!

Diese Aufgabe fokussiert auf drei grundlegenden Kompetenzen von Informatikerinnen und Informatikern.

Zuerst geht es darum, eine Lösung zu finden, die gegebene Einschränkungen einhält, oder nach Bedarf einen Lösungsvorschlag zu korrigieren.

Zweitens geht es um die Fähigkeit, Objekte über ihre Darstellung aus Teilinformationen rekonstruieren zu können. Das hängt mit der Generierung von Objekten (Objektdarstellungen) aus eingeschränkten verfügbaren Informationen zusammen, wenn man die Gesetzmässigkeit der Objekte kennt. Solche Vorgehensweisen kann man auch bei der Komprimierung anwenden.

Drittens kann man solche Baumfelder mit Schildern zur Erzeugung von selbst-verifizierenden Codierungen einsetzen. Vorkommende Fehler beim Eintragen oder beim Informationstransport können dann automatisiert erkannt oder sogar korrigiert werden.

Stichwörter und Webseiten

- Sudoku: `https://de.wikipedia.org/wiki/Sudoku`
- Fehlererkennung und Fehlerkorrektur:
  `https://de.wikipedia.org/wiki/Fehlerkorrekturverfahren`
- Rekonstruktion von Objekten aus Teilinformationen
- Überprüfung der Korrektheit von Datendarstellungen

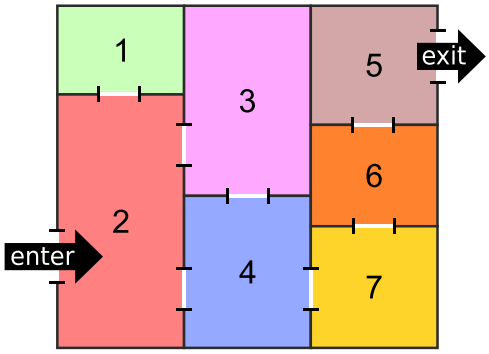
6. Museumsrundgang

Für ein neues Museum werden vier Grundrisse für die Räume vorgeschlagen. Jeder Grundriss enthält die sieben Räume 1 bis 7. Die Räume sollen so sein, dass die Besucher alle Räume besuchen können, ohne dabei einen Raum zweimal zu betreten.

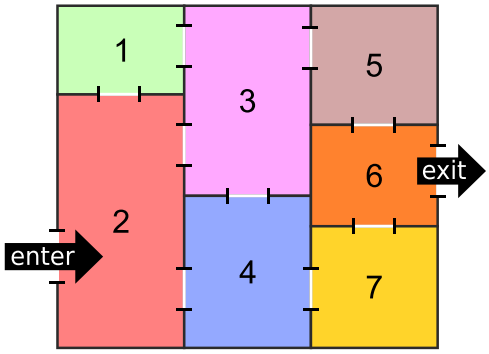
Die Besucher starten den Besuch bei «enter» und verlassen das Museum bei «exit».

Welcher Grundriss erlaubt es den Besuchern, jeden Raum genau einmal zu betreten und zu verlassen?

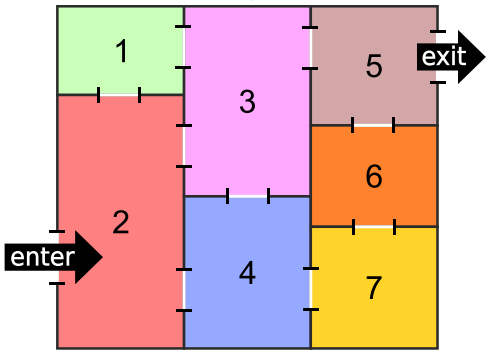
A)



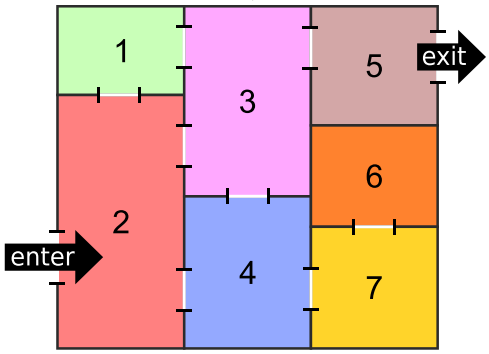
B)



C)

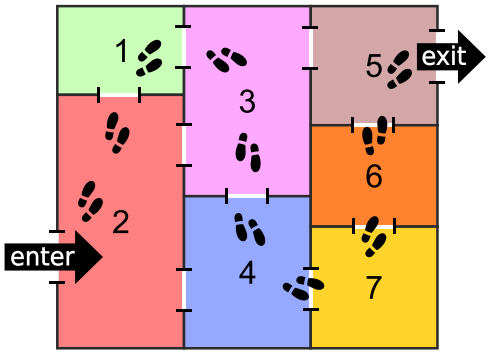


D)



Lösung

Nur der Grundriss C erlaubt es den Besuchern, jeden Raum genau einmal zu betreten und zu verlassen. Die Reihenfolge der Räume ist dabei: 2, 1, 3, 4, 7, 6, 5.



Generell ist eine solche immer Tour unmöglich, falls irgendeiner der Räume nur einen Eingang hat. Die Erklärung ist folgende: Falls ein Besucher diesen Raum betritt, muss er beim Verlassen dieses Raumes wieder zurück in den Raum, aus dem er gekommen ist, und verletzt dabei die Regel, dass er jeden Raum nur einmal betreten soll.

Im Grundriss A hat der Raum 1 nur einen Eingang.

Im Grundriss D hat der Raum 6 nur einen Eingang.

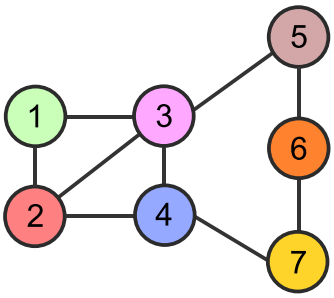
Im Grundriss B kann der letzte Raum 6 von Raum 5 oder von Raum 7 erreicht werden. Falls der Besucher vom Raum 5 kommt, kann er den Raum 7 betreten, aber danach nur durch den Raum 6 den Ausgang erreichen (oder umgekehrt), was die Regeln wieder verletzt.

Dies ist Informatik!

Die meisten Kinder oder Jugendliche lösen das Problem ohne zusätzliche abstrakte Darstellungen durch Probieren. Damit verwenden sie zu gewissem Grad die allgemeine Strategie des *Backtrackings*. Sie erkennen mindestens, dass man aus erfolglosen Versuchen lernen kann und in diesem Fall zurückgehen kann, um eine andere Möglichkeit auszuprobieren. Gleichzeitig ist man mit dem wichtigen Konzept des *Nichtdeterminismus* konfrontiert, weil man häufig mehrere Möglichkeiten zur Auswahl hat.

Die Aufgabe ist ein Beispiel für ein bekanntes Problem in der Informatik, der Suche nach einem *Hamiltonschen Weg*. In einer abstrakten Darstellung als *Graph* entspricht jeder Raum einem *Knoten* und jede Tür zwischen zwei Räumen einer *Kante* zwischen beiden entsprechenden Knoten.

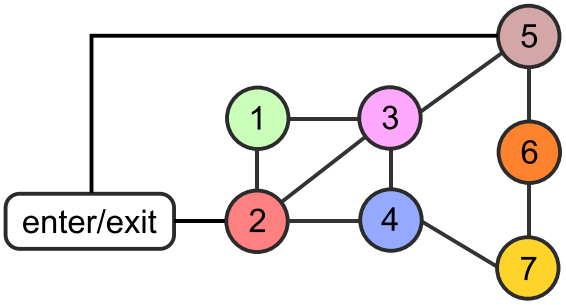
Abstrakt dargestellt sieht die Aufgabe folgendermassen aus:



Es geht jetzt darum, in diesem Graphen einen Weg mit folgenden Eigenschaften zu finden:

1. Der Weg startet in 2 («enter»).
2. Der Weg endet in 5 («exit»).
3. Jeder Knoten wird genau einmal besucht.

Falls man den Aussenraum zu einem Knoten zusammenfasst, dann entspricht das Ganze der Suche nach einem Hamiltonschen Kreis (einer Rundtour), wo auch jeder Knoten genau einmal durchlaufen wird und man am Ende dann aber wieder beim Anfangsknoten ist.



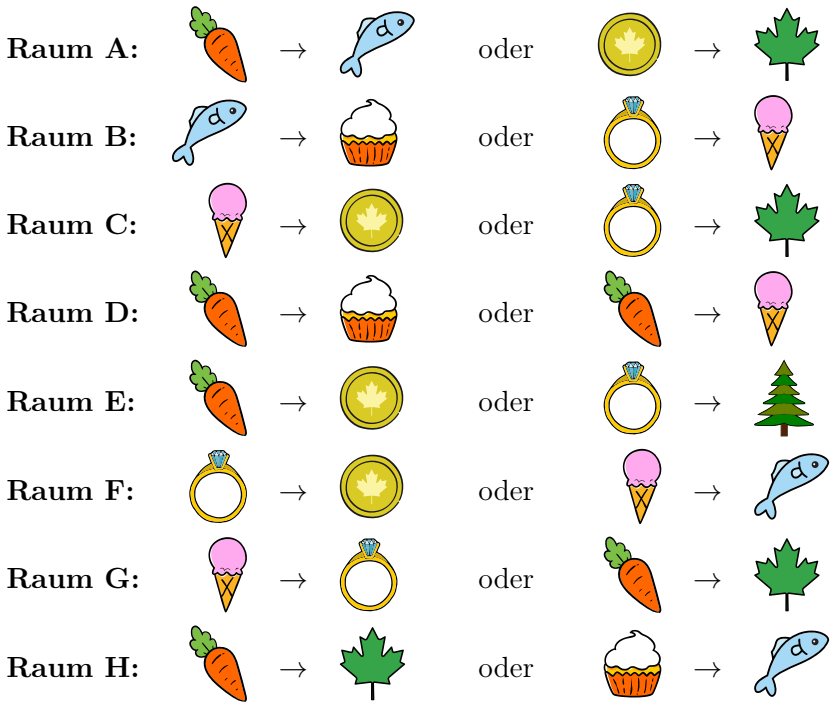
Stichwörter und Webseiten

- Graphentheorie, Knoten, Kante: `https://de.wikipedia.org/wiki/Graphentheorie`
- Hamiltonkreisproblem: `https://de.wikipedia.org/wiki/Hamiltonkreisproblem`

7. Biber im Schloss

Ein schlauer Biber braucht einen Tannenbaum um im Fluss einen Damm zu bauen. Leider hat er aber nur ein Rüebli . Im Schloss ist heute Markttag und der Biber will dort sein Rüebli gegen einen Tannenbaum eintauschen.

Jeder Raum des Schlosses bietet zwei Tauschangebote. Die Tabelle zeigt diese Angebote:



Zum Beispiel kann der Biber in Raum B für einen Ring ein Cornet bekommen, aber nicht umgekehrt.

In welcher Reihenfolge soll der schlaue Biber durch die Räume gehen, um letztlich den gewünschten Tannenbaum zu besitzen?

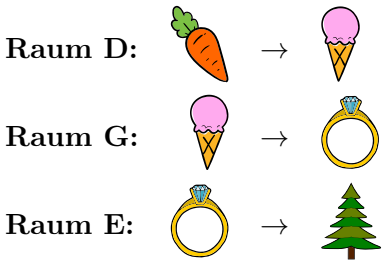
A) DGE: Zuerst Raum D, dann Raum G und zuletzt Raum E.
B) GGE: Zuerst Raum G, dann nochmal Raum G und zuletzt Raum E.
C) AGE: Zuerst Raum A, dann Raum G und zuletzt Raum E.
D) DBC: Zuerst Raum D, dann Raum B und zuletzt Raum C.

Lösung

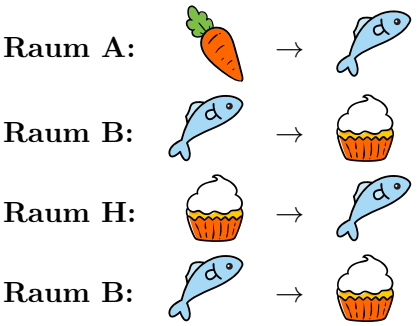
Die richtige Antwort ist A) DGE: Zuerst Raum D, dann Raum G und zuletzt Raum E.

Im Raum D tauscht der Biber sein Rüebli gegen ein Cornet. Danach geht er in Raum G, wo er das Cornet gegen einen Ring tauscht. Am Ende geht der Biber in den Raum E, um den Ring gegen einen Tannenbaum zu tauschen.

Diese Gesamtabfolge sieht so aus:



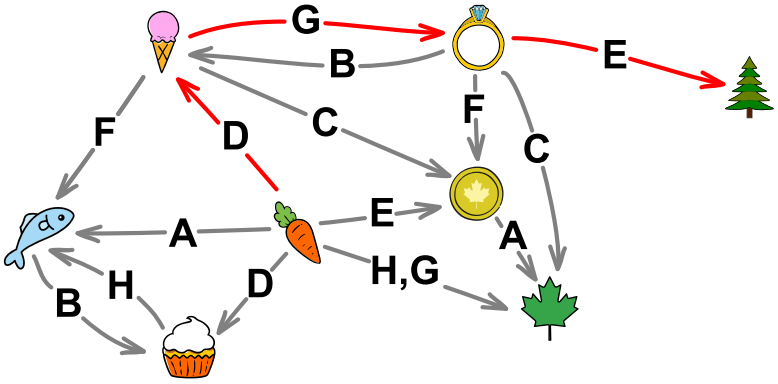
Um eine passende Reihenfolge der Räume zu finden, sind zwei unterschiedliche Strategien zielführend. Die erste Strategie versucht alle Tauschmöglichkeiten in Betracht zu ziehen. Sie beginnt damit, dass man beim ersten Tausch das Rüebli in fünf Räumen (A, D, E, G und H) gegen 6 verschiedene Objekte eintauschen kann. Danach werden für diese 6 Objekte wieder alle Eintauschmöglichkeiten betrachtet. Dies ist aufwendig und kann sogar im Kreis laufen, wie in folgendem Beispiel, wo der Biber beliebig oft die Räume B und H besuchen kann:



Somit ist diese erste Strategie sehr aufwendig und nur mit Glück in kurzer Zeit erfolgreich.

Die zweite Strategie führt in dieser konkreten Aufgabe schnell zum Ziel. Sie basiert darauf, die Suche vom gewünschten Ziel her, also dem Tannenbaum, zu beginnen. Nur im Raum E kann der Biber den gewünschten Tannenbaum bekommen. Den Tannenbaum bekommt man nur im Tausch gegen einen Ring. Das nächste gewünschte Objekt ist also ein Ring! Auch den Ring kann man nur in einem Raum bekommen, nämlich im Raum G im Tausch gegen ein Cornet. Ein Cornet erhält man entweder in Raum B gegen einen Ring oder in Raum D gegen ein Rüebli. Der schlaue Biber muss also seine Tauschvorgänge in Raum D beginnen.

Für eine bessere Übersicht kann die Tabelle der möglichen Tauschvorgänge als Graph mit gerichteten Kanten (Pfeilen) dargestellt werden. Jeder Knoten des Graphen steht für ein Tauschobjekt und jede ausgehende Kante steht für eine Tauschmöglichkeit. Zusätzlich steht auf der Kante, in welchem Raum diese Tauschmöglichkeit besteht.



Diese visuelle Darstellung der Tauschobjekte, Tauschmöglichkeiten und Räume erlaubt es, leicht herauszufinden, wie man vom Rüebli zum Tannenbaum kommt, nämlich auf einem Weg im gerichteten Graphen: Zuerst Raum D, dann Raum G und zuletzt Raum E.

Dies ist Informatik!

Berechnungsprozesse kann man auf einer allgemeinen Ebene betrachten als *Folgen von Transformationen* (hier sind es Tauschvorgänge) oder gleichwertig als *Folgen von Zuständen* eines Systems. Der Startzustand des Systems ist in unserem Fall das Rüebli und die Transformation (der *Berechnungsschritt*) vom Rüebli zum Cornet ändert diesen Zustand zum Cornet.

Ein Berechnungsschritt führt also von einem Zustand zu einem anderen. Eine Abfolge von Berechnungsschritten nennt man auch eine Berechnung.

Somit behandelt diese Aufgabe auch Berechnungen auf einer sehr allgemeinen Ebene. Das System ist im vorliegenden Fall *nichtdeterministisch*; das bedeutet, dass es manchmal mehrere mögliche Berechnungsschritte gibt, also wie in der Aufgabe mehrere Tauschmöglichkeiten. Nichtdeterminismus ist ein weiteres wichtiges Konzept der Modellierung in der Informatik. Die möglichen Berechnungsschritte werden durch *Transformationsregeln* (die Tabelle mit Tauschmöglichkeiten) beschrieben. Zu bestimmen, ob der Biber ein Rüebli in einen Tannenbaum eintauschen kann, also ob ein bestimmter *Zielzustand* des Systems von einem bestimmten *Startzustand* aus erreichbar ist, ist das berühmte *Erreichbarkeitsproblem* mit zahlreichen Anwendungen.

Die Aufgabe oben zeigt, dass es manchmal eine gute Idee ist, vom Zielzustand her den Startzustand zu suchen statt umgekehrt. Diese Strategie nennt man auch *Rückwärtssuche*.

Beim Vergleich der verschiedenen Lösungsstrategien erkennt man, dass der gerichtete Graph eine anschauliche Möglichkeit der Darstellung eines sogenannten *Zustandsraumes* eines Systems mit allen möglichen Übergängen zwischen Zuständen ist. In diesem Basismodell könnte man die bekannten grundlegenden *Suchalgorithmen* in Graphen, nämlich *Breitensuche* und *Tiefensuche* ansprechen.

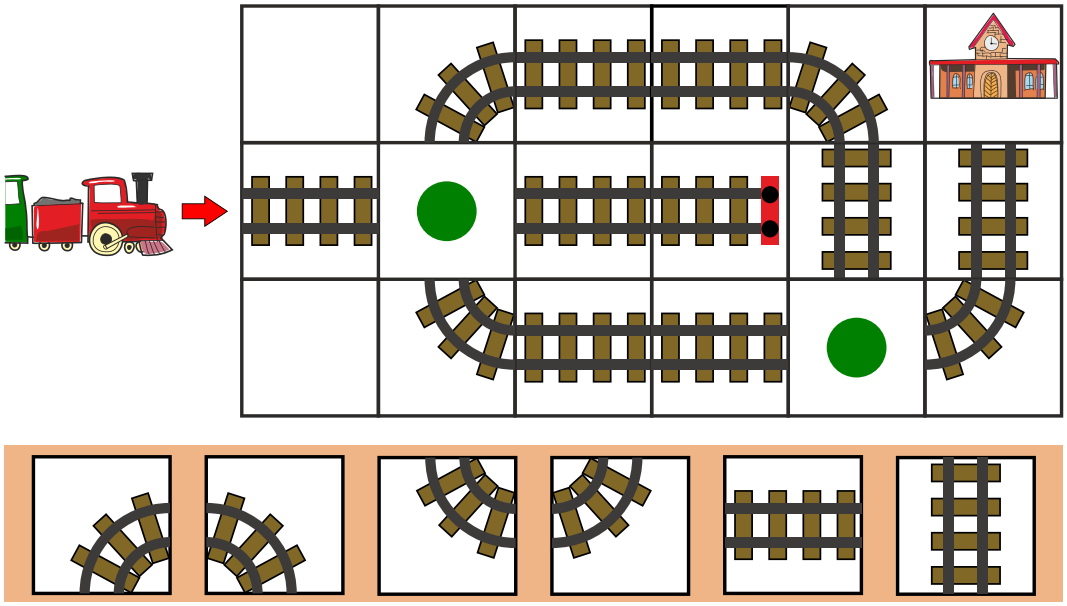
Stichwörter und Webseiten

- Graphentheorie: `https://de.wikipedia.org/wiki/Graph_(Graphentheorie)`

- Erreichbarkeitsproblem:
  `https://de.wikipedia.org/wiki/Erreichbarkeitsproblem_in_Graphen`
- Tiefensuche: `https://de.wikipedia.org/wiki/Tiefensuche`
- Breitensuche: `https://de.wikipedia.org/wiki/Breitensuche`

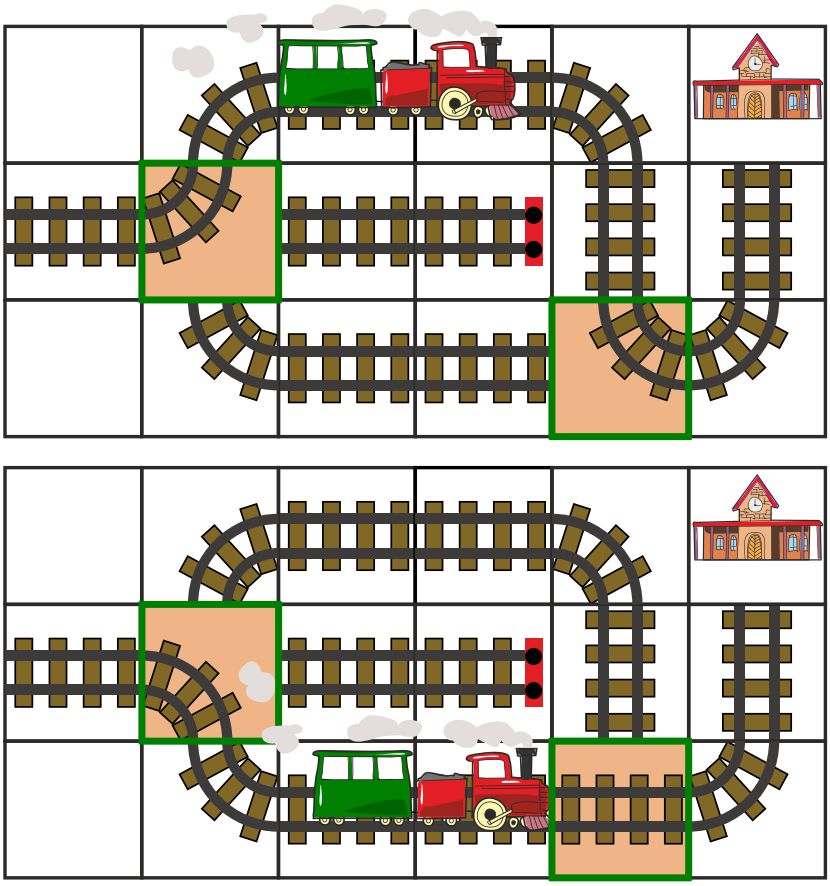
8. Nächster Halt, Bahnhof!

Lege Schienen auf die grünen Punkte, so dass der Zug zum Bahnhof fahren kann.



Lösung

Für dieses Problem gibt es folgende 2 Lösungen:



Bei anderen Kombinationen entgleist der Zug oder fährt gegen den Prellbock.

Dies ist Informatik!

Wie ein Zug stur entlang den Schienenstücken fährt, führt ein Computer stur die Anweisungen eines Programms aus. Er kann nicht erkennen, wenn das Programm einen Fehler enthält und kann dann «abstürzen», so wie ein Zug entgleisen kann, wenn die Schienen falsch gebaut wurden. Beim Schreiben eines Programms muss man also viel sorgfältiger sein, als wenn man beispielsweise einer Person den Weg zum Bahnhof erklärt.

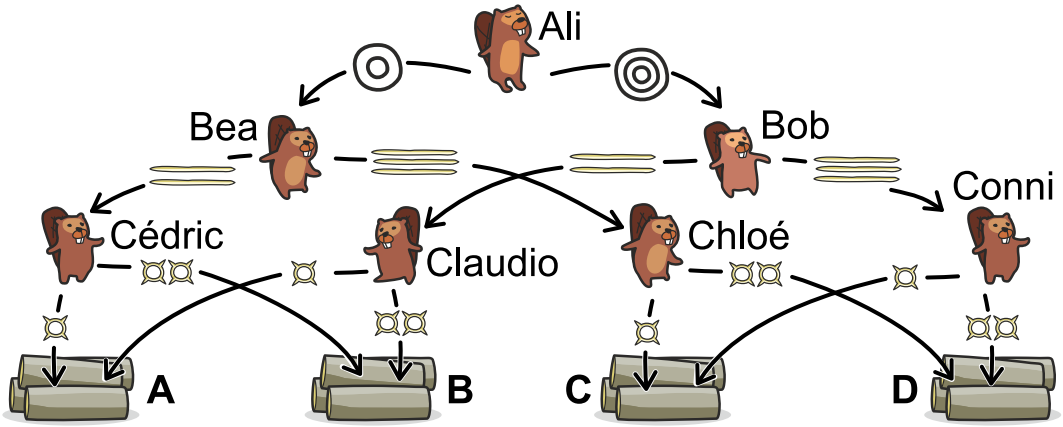
In der Aufgabe oben geht es darum, in einem Programm an den richtigen Stellen fehlende Befehle einzufügen, sodass die Zielsetzung erreicht wird.

Stichwörter und Webseiten

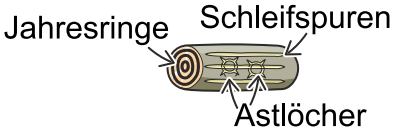
- Programm
- Anweisung: `https://de.wikipedia.org/wiki/Anweisung_(Programmierung)`
- `https://de.wikipedia.org/wiki/Algorithmus`

9. Baumstämme auf Stapel

Im Biberdorf werden die Stämme nach drei Eigenschaften (Anzahl Jahresringe, Anzahl Schleifspuren in der Rinde und Anzahl der Astlöcher) in vier Gruppen (A, B, C, D) verteilt. Wie das abläuft, zeigt das Entscheidungsdiagramm.



Beispielsweise wird dieser Stamm aufgrund folgender Entscheidungen auf den Stapel D gelegt:



- Ali sieht drei Jahresringe und gibt den Stamm an Bob.
- Bob sieht drei Schleifspuren und gibt den Stamm an Conni.
- Conni sieht zwei Astlöcher und legt den Stamm auf den Stapel D.

Auf welchem Stapel wird dieser Stamm abgelegt?

A) Stapel A
B) Stapel B
C) Stapel C
D) Stapel D

Lösung

Die korrekte Antwort ist Stapel C. Dies ist so, weil Ali zwei Jahresringe sieht und den Stamm an Bea gibt. Bea sieht drei Schleifspuren und gibt den Stamm an Chloé weiter. Chloé sieht ein Astloch und legt den Stamm auf den Stapel C.

Wenn man will, kann man für jeden Stapel bestimmen, welche Stämme auf den jeweiligen Stapel gehören. Auf jedem Stapel gibt zwei Arten von Stämmen.

Auf Stapel **A**:

- Stämme mit 2 Jahresringen, 2 Schleifspuren und 1 Astloch.
- Stämme mit 3 Jahresringen, 2 Schleifspuren und 1 Astloch.

Auf Stapel **B**:

- Stämme mit 2 Jahresringen, 2 Schleifspuren und 2 Astlöchern
- Stämme mit 3 Jahresringen, 2 Schleifspuren und 2 Astlöchern

Auf Stapel **C**:

- Stämme mit 2 Jahresringen, 3 Schleifspuren und 1 Astloch
- Stämme mit 3 Jahresringen, 3 Schleifspuren und 1 Astloch

Auf Stapel **D**:

- Stämme mit 2 Jahresringen, 3 Schleifspuren und 2 Astlöchern
- Stämme mit 3 Jahresringen, 3 Schleifspuren und 2 Astlöchern

Dies ist Informatik!

Diese Aufgabe berührt mehrere Konzepte der Informatik.

Vor allem wird das Konzept der *Entscheidungsdiagramme* angesprochen, die sehr vielseitige Anwendungen in der Informatik haben. Hier verwendet man sie zur *Klassifizierung* von Objekten in gewählte Kategorien. (Sehr häufig sind es *Entscheidungsbäume*, eine spezielle Art von Entscheidungsdiagrammen. Das Entscheidungsdiagramm der Aufgabe ist hier kein Entscheidungsbaum, weil auf der untersten Ebene je zwei Gruppen auf denselben Stapel gelegt werden.)

Man kann das Entscheidungsdiagramm hier auch als eine abstrakte Darstellung der Werte einer Funktion von mehreren Variablen ansehen. Terminologisch genau spricht man in der Informatik von «branching programs».

Zudem spricht man hier auch das Konzept der *Attribute* (Merkmale oder Eigenschaften) von Objekten an. Hier haben die Objekte drei Attribute (Jahresringe, Schleifspuren, Astlöcher), wobei jedes Attribut zwei mögliche Werte hat (zwei oder drei Jahresringe oder Schleifspuren und ein oder zwei Astlöcher).

Es gibt viele Anwendungsmöglichkeiten für solche Entscheidungsdiagramme. Eine davon ist die Klassifizierung von Paketen beim Versenden durch ein Netzwerk (mit Routern oder Switches).

Stichwörter und Webseiten

- Entscheidungsbaum: `https://de.wikipedia.org/wiki/Entscheidungsbaum`
- Klassifizierung

A. Aufgabenautoren

Serge Adam

Wilfried Baumann

Carlo Bellettini

Linda Björk Bergsveinsdóttir

Daniela Bezáková

Lucia Budinská

Sarah Chan

Marios O. Choudary

Valentina Dagienė

Christian Datzko

Susanne Datzko

Lidia Feklistova

Fabian Frei

Husnul Hakim

Juraj Hromkovič

Alisher Ikramov

Ungyeol Jung

Vaidotas Kinčius

Regula Lacher

Vu Van Luan

Hamed Mohebbi

Kwangsik Moon

Xavier Muñoz

Tom Naughton

Gabriel Parriaux

Jean-Philippe Pellet

Margot Phillipps

Wolfgang Pohl

Pedro Ribeiro

Peter Rossmanith

Vipul Shah

Peter Tomcsányi

Monika Tomcsányiová

Jiří Vaníček

Michael Weigend

Jonas Winckler

Michal Winczer

B. Sponsoring: Wettbewerb 2020

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

http://www.verkehrshaus.ch/

Standortförderung beim Amt für Wirtschaft und Arbeit Kanton Zürich

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

http://www.ubs.com/
Wealth Management IT and UBS Switzerland IT

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

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

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

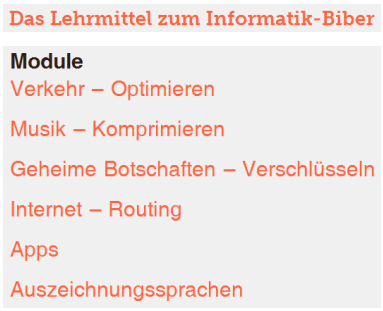
http://www.phlu.ch/
Pädagogische Hochschule Luzern

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

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



`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.

Werden Sie SVIA Mitglied – `http://svia-ssie-ssii.ch/svia/mitgliedschaft` und unterstützten 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.