INFORMATIK-BIBER SCHWEIZ
CASTOR INFORMATIQUE SUISSE
CASTORO INFORMATICO SVIZZERA

Aufgaben und Lösungen 2018
Schuljahre 9/10

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

Herausgeber:
Christian Datzko, Susanne Datzko, Hanspeter Erni

AEIOUHWY	X
BFPV	1
CGJKQSXZ	2
DT	3
L	4
NM	5
R	6

SVIA
www.svia-ssie-ssii.ch
schweizerischerverein für informatikind erausbildung//société suisse pour l'infor matiquedans l'enseignement//società sviz zera per l'informatica nell'insegnamento

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 Trachsler.

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 19. Januar 2019 mit dem Textsatzsystem LaTeX erstellt.

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“ 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

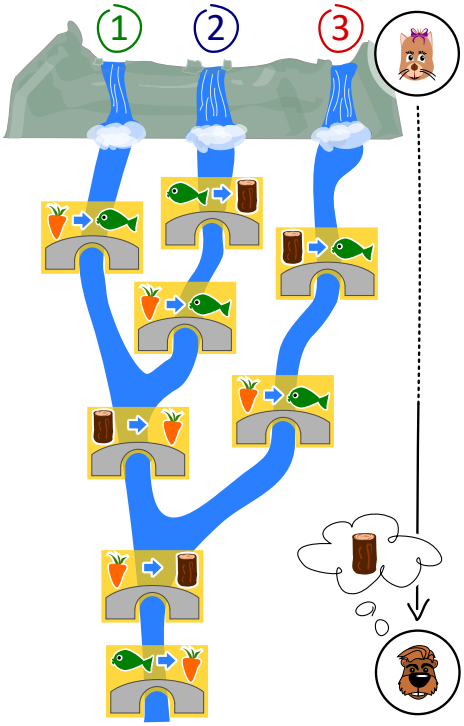
Das international angewandte System zur Punkteverteilung soll dem erfolgreichen Erraten der richtigen Lösung durch die Teilnehmenden entgegenwirken.
Jede Teilnehmerin und jeder Teilnehmer hatte zu Beginn 45 Punkte („Kleiner Biber“: 27 Punkte, Stufen 5 und 6: 36 Punkte) auf dem Punktekonto.
Damit waren 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:

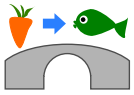
SVIA-SSIE-SSII Schweizerischer Verein für Informatik in der Ausbildung
Informatik-Biber
Hanspeter Erni
`https://www.informatik-biber.ch/de/kontaktieren/`
`https://www.informatik-biber.ch/`
 `https://www.facebook.com/informatikbiberch`

Inhaltsverzeichnis

1. Wasserfälle



Katja sitzt oben auf dem Berg. Der Berg hat drei Wasserfälle. Die Wasserfälle fliessen in einen Fluss.
Katja kann entweder ein Rüebli oder einen Fisch in einen der Wasserfälle fallen lassen. Der Fluss hat mehrere Brücken mit Trollen. Die Trolle ersetzen Gegenstände, die unter den Brücken durchgehen.



Wenn zum Beispiel ein Rüebli unter einer solchen Brücke durchschwimmt, ersetzen die Trolle das Rüebli 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 Rüebli in den Wasserfall 2 fallen.

D) Sie lässt ein Rüebli in den Wasserfall 3 fallen.

Lösung

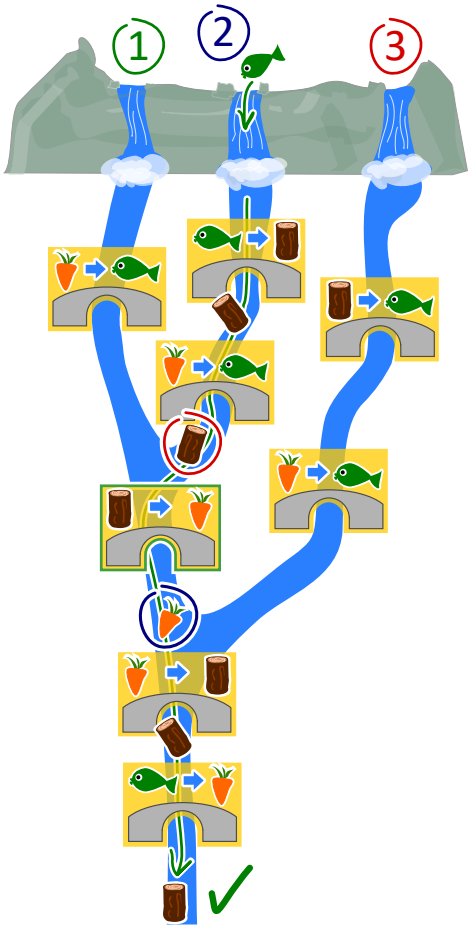
Die richtige Antwort ist: B) Sie lässt einen Fisch in den Wasserfall 2 fallen.
Was passiert bei den verschiedenen Lösungsvorschlägen:

A) Ein in den Wasserfall 1 geworfener Fisch wird nur unter der letzten Brücke gewechselt. So erhält Justus ein Rüebli.

B) Ein in den Wasserfall 2 geworfener Fisch wird in Holz, dann Rüebli und dann wieder in Holz umgewandelt. So erhält Justus Holz. Das ist somit korrekt.

C) Ein Rüebli, die in den Wasserfall 2 fallen gelassen wird, wird in Fisch und dann Rüebli geändert. So bekommt Justus ein Rüebli.

D) Ein Rüebli, die in den Wasserfall 3 fällt, wird in Fisch und dann in Rüebli verwandelt. So bekommt Justus ein Rüebli.



Ein anderer Lösungsansatz für diese Aufgabe besteht darin, rückwärts zu beginnen:
Um am Ende Holz zu bekommen, muss der gefallene Gegenstand ein Rüebli sein, wenn er die vorletzte Brücke passiert. Die einzige Möglichkeit, an diesem Punkt ein Rüebli 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.

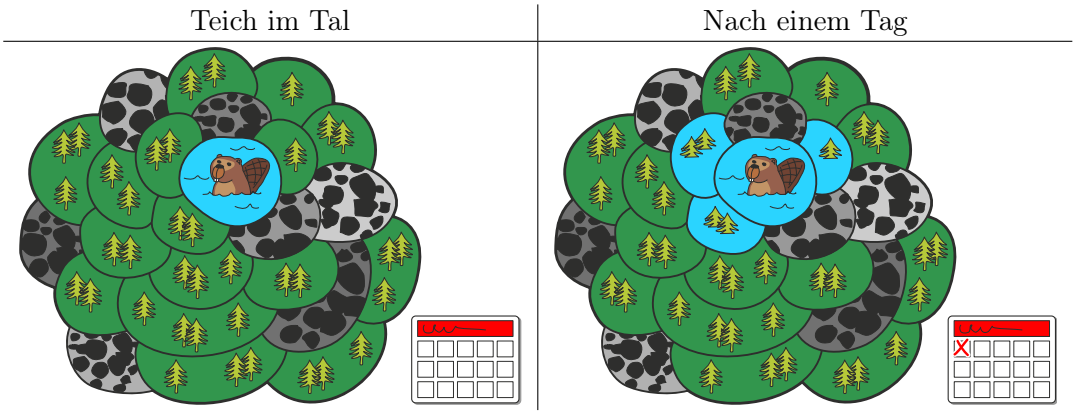
Stichwörter und Webseiten

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`

2. Biber-Teich

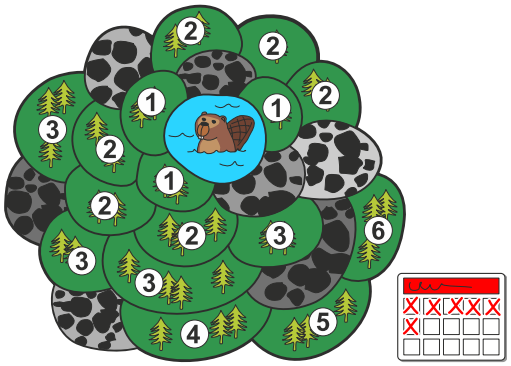
In einem Tal liegt ein kleiner Teich. Er ist umgeben von Landstücken mit Wald oder mit Felsen. Im Teich leben einige Biber.
Eines Tages wird den Bibern der Teich zu klein und nun fluten sie den Wald. An jedem Tag fluten sie alle Waldstücke, die an bereits geflutete Waldstücke angrenzen. Nach einem Tag 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

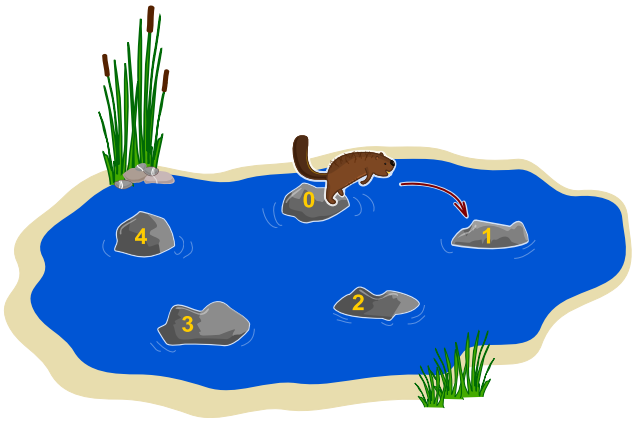
Wavefront Algorithm, Breadth-First Search

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

3. 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:

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



Der stärkste Biber springt heute ganze 129 *mal. Er startet auf Feld* 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/Schriftliche_Division`
- `https://en.wikipedia.org/wiki/Euclidean_division`
- `https://de.wikipedia.org/wiki/RSA-Kryptosystem`

4. 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

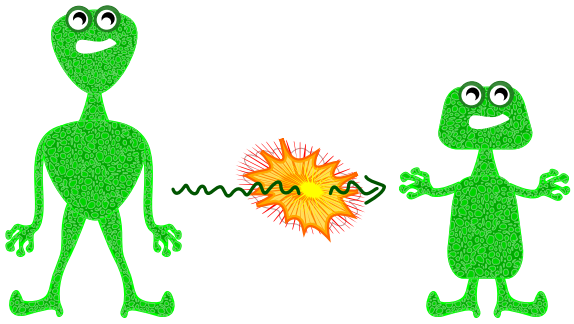
- `https://de.wikipedia.org/wiki/Binärer_Suchbaum`
- `https://de.wikipedia.org/wiki/AVL-Baum`

5. Aliens!

Ein Alien besteht aus einem Kopf, einem Rumpf, zwei Armen und zwei Beinen. Ein Alien kann durch folgende Befehle verändert werden, dabei ist es auch möglich, dass ein Körperteil mehrfach verändert wird.

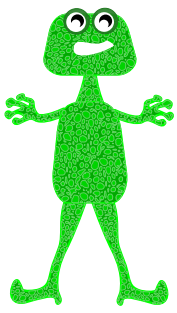
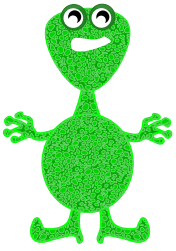
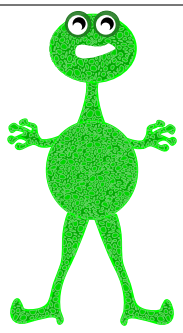
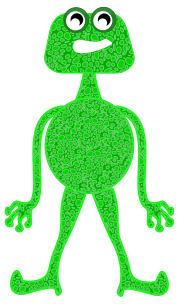
$K(r)$	Der Kopf wird rund.	
$K(4)$	Der Kopf wird viereckig.	
$K(3)$	Der Kopf wird dreieckig.	
$R(r)$	Der Rumpf wird rund.	
$R(4)$	Der Rumpf wird viereckig.	
$R(3)$	Der Rumpf wird dreieckig.	
$A(+)$	Die Arme werden lang.	
$A(-)$	Die Arme werden kurz.	
$B(+)$	Die Beine werden lang.	
$B(-)$	Die Beine werden kurz.	

Die einzelnen Befehle werden von links nach rechts ausgeführt. Zum Beispiel ergibt $K(r)$, $R(4)$, $K(4)$, $A(-)$, $B(-)$ das folgende Alien:

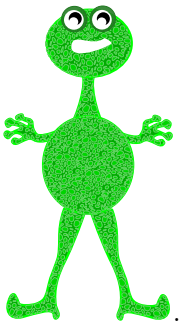


Wie schaut das Alien nach den folgenden Befehlen aus?

$K(3)$, $B(+)$, $R(3)$, $A(+)$, $K(r)$, $A(-)$, $R(r)$

A)	B)	C)	D)
			

Lösung

Die richtige Antwort ist C)  .
Für jeden Körperteil gilt nur der letzte Befehl. Ein vorhergehender Befehl für den selben Körperteil ist in der Endform des Aliens nicht erkennbar, da ein folgender Befehl die Veränderung wirkungslos macht.
Daher ist das Ergebnis ein Alien mit einem runden Rumpf ($R(r)$), kurzen Armen ($A(-)$), einem runden Kopf ($K(r)$) und langen Beinen ($B(+)$). Die anderen Aliens unterscheiden sich von der Lösung C) in mindestens zwei Eigenschaften, so dass sie offensichtlich falsch sind.

Dies ist Informatik!

Beim Ausführen eines Programms werden die Befehle der Reihe nach abgearbeitet. Nachfolgende Befehle können dabei die Wirkung vorhergehender Befehle wieder verändern.
In Computerprogrammen passiert das häufig bei Variablen, in denen Werte gespeichert werden, die sich während der Laufzeit des Programms mehrfach verändern. Man kann daher die Form der vier Körperteile jeweils wie eine Variable ansehen, in der die aktuelle Form gespeichert ist. Der Befehl „$K(r)$“ bewirkt dann, dass in der Variablen für die Kopfform der neue Wert „r“ gespeichert wird.
Die Notation „$K(r)$“ ist funktional gedacht. Man ruft die Funktion „$K()$“ auf und gibt ihr das Argument „r“ mit. Dies wird häufig verwendet, da die Funktion „$K()$“ nebenbei noch prüfen kann, ob das mitgegebene Argument überhaupt gültig ist. Wenn das nicht nötig ist oder wenn die Variable nur lokal verwendet wird, kann man sie auch direkt überschreiben, hierfür verwendet man häufig den Zuweisungsoperator „=“. Man würde dann beispielsweise „$K = r$“ im Programm schreiben.

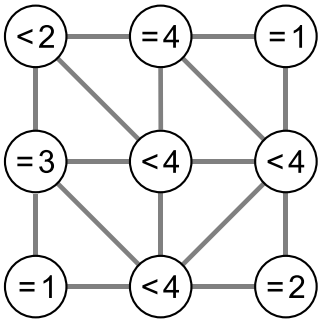
Stichwörter und Webseiten

Variable, Sequenz

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

6. Nachbarn

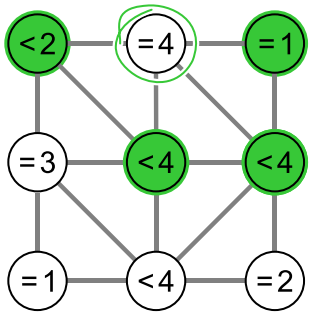
Im Bild unten sind neun Kreise zu sehen, die teilweise miteinander verbunden sind. Eine Verbindung macht sie zu Nachbarn. Durch Anklicken können Kreise ausgewählt werden. Ein ausgewählter Kreis ist grün dargestellt, ein nicht ausgewählter Kreis weiss.
In jedem Kreis steht ein Ausdruck, der anzeigt, wie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



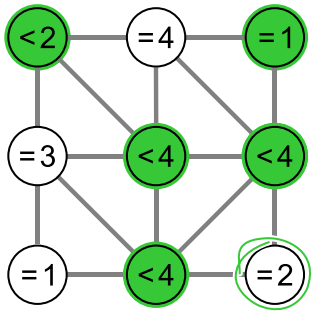
Wähle die Kreise so aus, dass die Bedingungen in allen neun Kreisen gleichzeitig erfüllt sind.

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/Heuristik#Informatik`
- `https://de.wikipedia.org/wiki/Brute-Force-Methode`

7. 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.

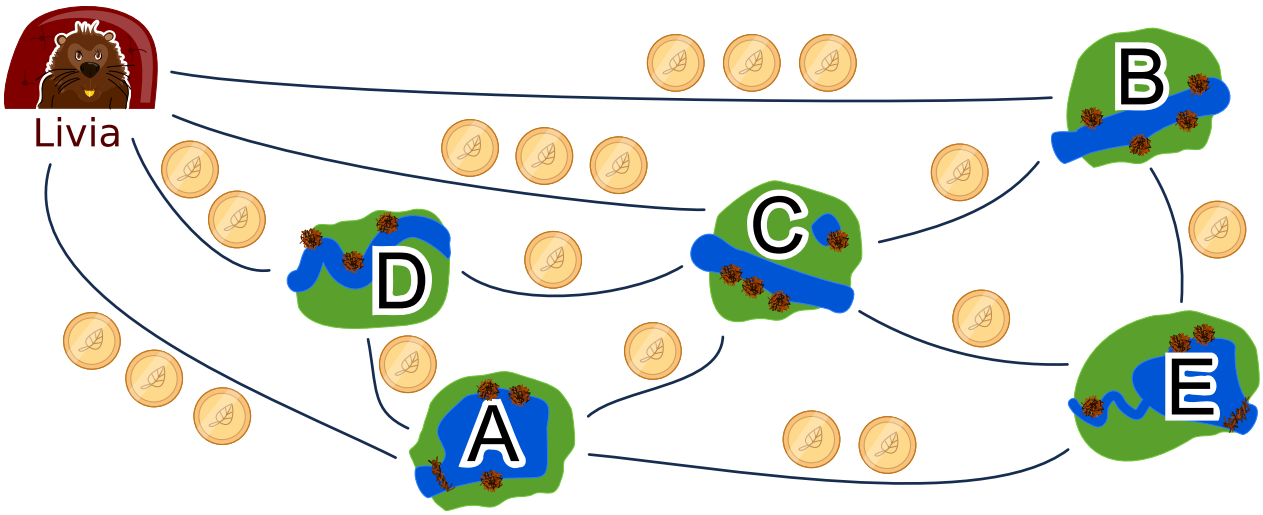
Stichwörter und Webseiten

Analyse, Verifizierung und Validierung von Software

- `https://en.wikipedia.org/wiki/Software_verification`
- `https://de.wikipedia.org/wiki/Verifizierung_und_Validierung`

8. 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:

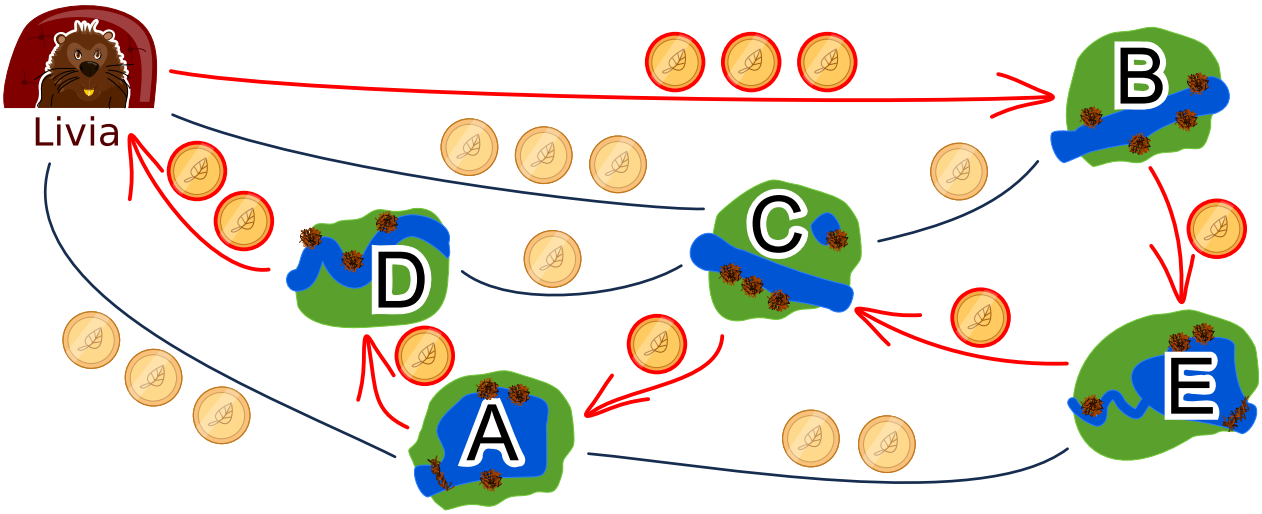
$$\text{Start} \to \text{B} \to \text{E} \to \text{A} \to \text{D} \to \text{C} \to \text{Start}.$$

Dieser Weg kostet $3 + 1 + 2 + 1 + 1 + 3 = 11$ Bibermünzen.
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 Bibermünzen. Es gibt keine bessere Lösung, denn von Livias Zuhause aus kannst du einmal den Zwei-Bibermünzen-Weg und dann einen Drei-Bibermünzen-Weg gehen. Zu den vier weiteren Knoten gehören vier Wege, die jeweils mindestens eine Bibermünze kosten, was bereits 9 Bibermünzen ergibt.
Alle anderen Lösungen kosten mehr:

- Kosten von 10 Bibermünzen: Start → A → D → C → E → B → Start
- Kosten von 10 Bibermünzen: Start → A → E → B → C → D → Start
- Kosten von 10 Bibermünzen: Start → B → C → E → A → D → Start
- Kosten von 10 Bibermünzen: Start → B → E → A → C → D → Start
- Kosten von 10 Bibermünzen: Start → B → E → C → D → A → Start
- Kosten von 10 Bibermünzen: Start → C → B → E → A → D → Start
- Kosten von 10 Bibermünzen: Start → D → A → E → B → C → Start
- Kosten von 10 Bibermünzen: Start → D → A → E → C → B → Start
- Kosten von 10 Bibermünzen: Start → D → C → A → E → B → Start
- Kosten von 10 Bibermünzen: Start → D → C → B → E → A → Start
- Kosten von 11 Bibermünzen: Start → B → E → A → D → C → Start
- Kosten von 11 Bibermünzen: 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 Bibermünzen 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.

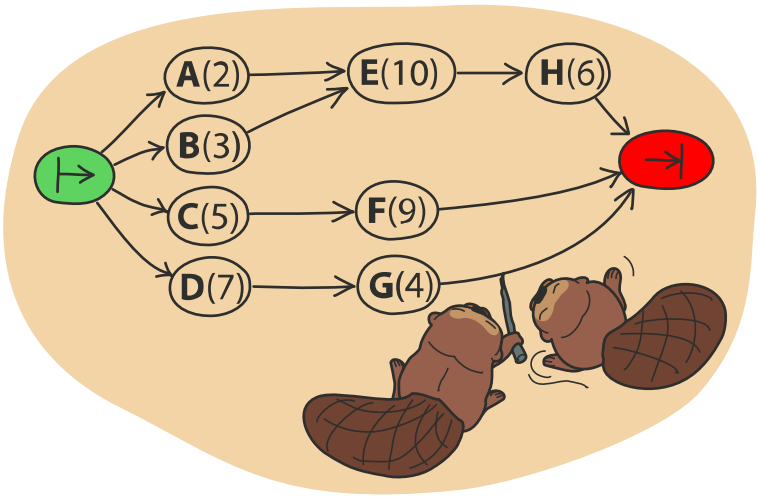
Stichwörter und Webseiten

Optimierung, Problem eines Handlungsreisenden

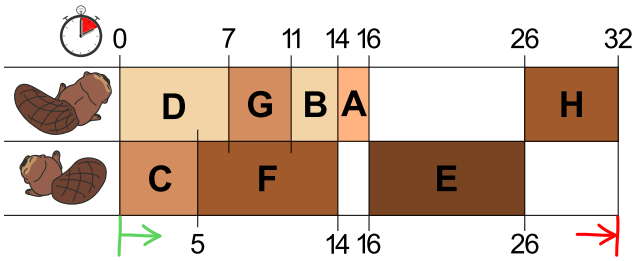
- `https://de.wikipedia.org/wiki/Problem_des_Handlungsreisenden`
- `https://de.wikipedia.org/wiki/Optimierungsproblem`

9. 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.



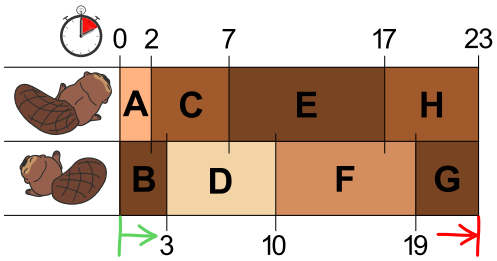
Die Abbildung unten zeigt einen möglichen Arbeitsplan der beiden Biber, der 32 Stunden benötigt. Es geht aber schneller!



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

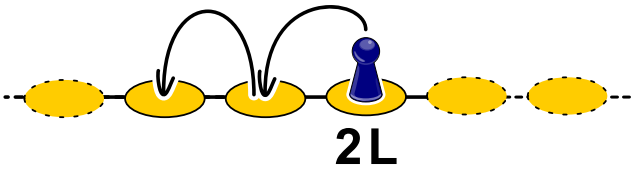
Scheduling, Greedy-Algorithmus

- `https://de.wikipedia.org/wiki/Scheduling`
- `https://de.wikipedia.org/wiki/Topologische_Sortierung`
- `https://de.wikipedia.org/wiki/Greedy-Algorithmus`

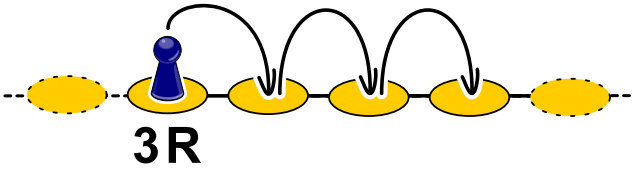
10. Hüpfspiel

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

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

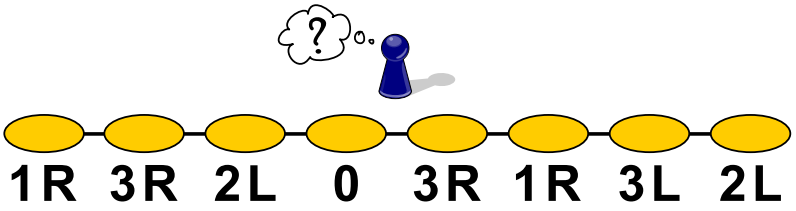


- nR: n Felder nach rechts hüpfen, 3R bedeutet also, drei Felder nach zu rechts hüpfen:



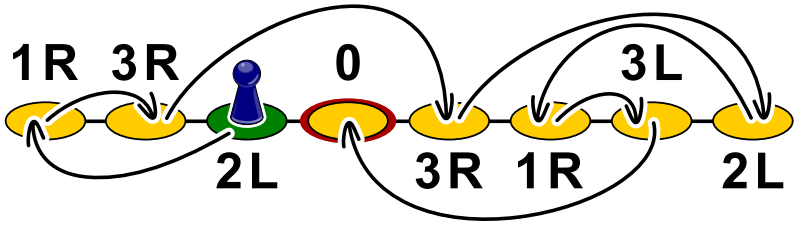
- 0: nicht mehr weiter hüpfen.

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



Lösung

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



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

Dies ist Informatik!

In der Informatik funktioniert das Speicherprinzip *„Verkettete Liste“* ähnlich wie die Hüpffelder: im Arbeitsspeicher werden Objekte mit Verweisen auf die Speicheradresse des nächsten Objekts gespeichert. So kann die Speicherverwaltung ein einzelnes Objekt beliebig im Arbeitsspeicher hinterlegen und muss nicht zeitaufwendig für alle Objekte einen zusammenhängenden Platz im Arbeitsspeicher erstellen. Der Programmierer wiederum muss sich um nichts kümmern, ausser dass er bei der Speicherverwaltung einen Speicher in der entsprechenden Grösse reserviert.
Was passiert aber, wenn Objekte im Speicher nicht mehr gebraucht werden? Während sich früher die Programmierer darum kümmern mussten, dass der Speicher wieder freigegeben wird (was leider häufig schief ging, so dass Programme mit der Zeit immer mehr Speicher verschwendeten und dann irgendwann aus Speichermangel abstürzten), haben moderne Programmiersprachen hierfür eine Art Müllabfuhr, die *„Garbage Collection“*, die regelmässig überprüft, ob Objekte im Speicher noch von anderen Objekten referenziert werden (ob es also noch Verweise auf sie gibt). Da manchmal grosse Strukturen von Objekten nicht mehr referenziert werden, muss genau wie im Beispiel zurückverfolgt werden, von welchem „Startfeld“ aus auf die anderen Objekte verwiesen wird.

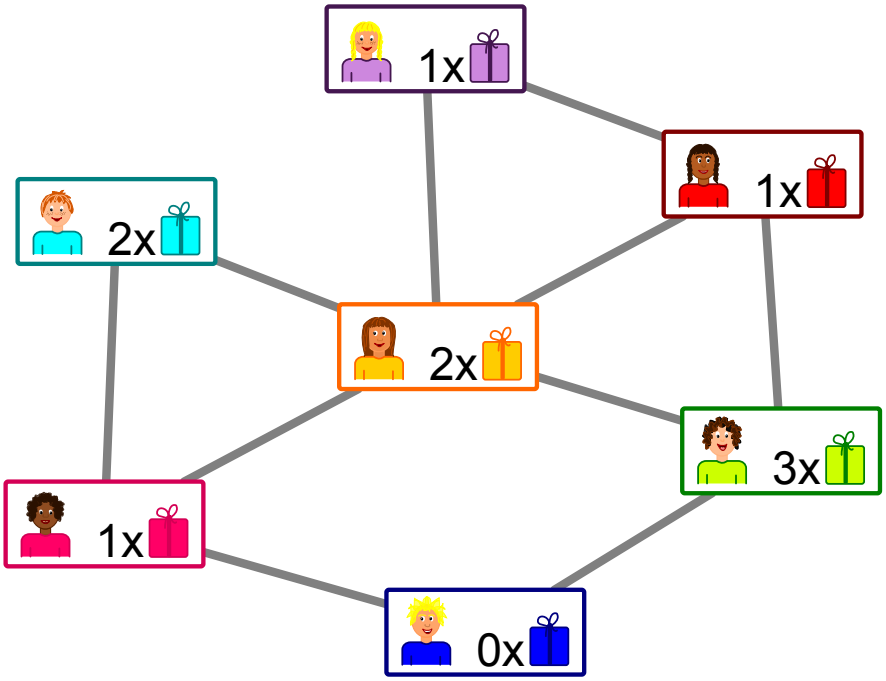
Stichwörter und Webseiten

Schlange, Speicherverwaltung, GOTO

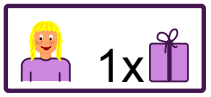
- `https://de.wikipedia.org/wiki/Garbage_Collection`
- `https://de.wikipedia.org/wiki/Erreichbarkeitsproblem_in_Graphen`

11. Geschenke

Das Bild zeigt die Freundschaften zwischen den Kindern in einem Haus. Eine Linie zwischen zwei Freunden bedeutet: Diese Kinder sind Freunde.



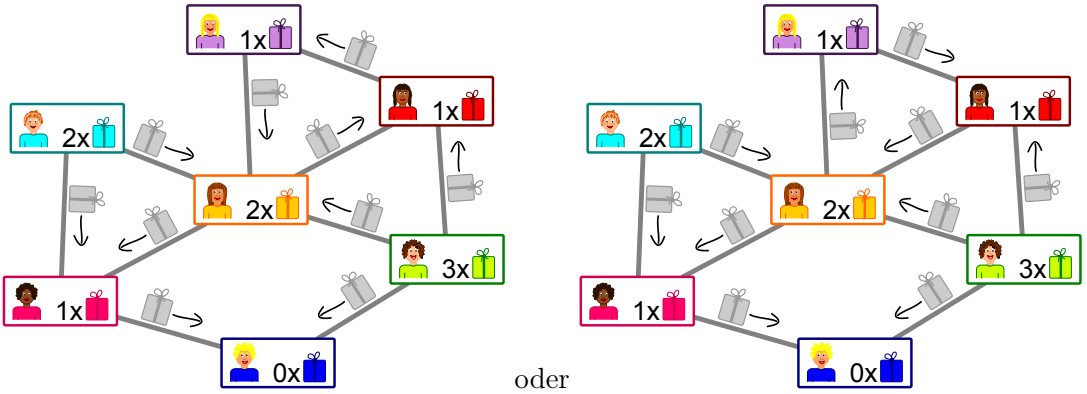
Die Hausbewohner planen ein Kinderfest mit Geschenken. Bei allen Paaren von Freunden soll ein Kind dem anderen Kind ein Geschenk besorgen.

Im Bild steht, wie viele Geschenke das Kind besorgen kann: 

bedeutet, dass das Kind ein Geschenk besorgen kann.
Entscheide für jedes Freundespaar, wer das Geschenk besorgt. Dabei soll kein Kind mehr Geschenke besorgen müssen, als es besorgen kann.

Lösung

Es gibt zwei Möglichkeiten, wie man für alle Paare von Freunden die „Schenkrichtung“ angeben kann, ohne dass ein Kind mehr Geschenke besorgen muss, als es besorgen kann.



Es lohnt sich bei dem Kind unten anzufangen: Es kann kein Geschenk besorgen und erhält deshalb je ein Geschenk von seinen Freundinnen links und rechts. Damit hat die Freundin links ihr Geschenk vergeben und erhält selbst je ein Geschenk von den anderen beiden Kindern, mit denen sie befreundet ist. Für die anderen Freundespaare sind die „Schenkrichtungen“ also klar.
Die einzige Auswahlmöglichkeit, die es noch gibt, ist, ob bei den drei Kindern oben rechts im Uhrzeigersinn oder gegen den Uhrzeigersinn geschenkt wird.

Dies ist Informatik!

Die Freundschaften zwischen den Kindern bilden ein Netzwerk aus Knoten (die Kinder) und Kanten (die Freundschaftsbeziehungen). Ähnlich sind „soziale Netzwerke“ aufgebaut mit Millionen von Benutzenden. Es gibt jedoch einen Punkt, in dem sich diese Systeme grundlegend unterscheiden können: In manchen gibt es wechselseitige „Freundschaften“, in denen die Verbindungen keine Richtung haben, so wie in dieser Aufgabe. In anderen gibt es „Anhänger“ (engl. “follower”), so dass die Verbindungen eine Richtung haben: Wenn du beispielsweise einer berühmten anderen Nutzerin „folgst“, muss sie nicht unbedingt auch dir folgen.
In dieser Aufgabe sollen die Freundschafts-Verbindungen „Schenkrichtungen“ bekommen. Das ist ein neuer Aspekt, denn die „Schenk-Kapazitäten“ der Kinder sind begrenzt und setzen so indirekt der Wahl der Richtungen Grenzen. Das Ziel ist, in jeder Freundschaft ein Geschenk zu schenken, ohne dass die Schenk-Kapazität eines Kindes überschritten wird. In der Informatik gibt es ähnliche Probleme: In einem Netzwerk (etwa die Kabel, die das Internet bilden) sind die Kapazitäten der Verbindungen begrenzt. Innerhalb dieser Grenzen aber ist es am besten, wenn diese Kapazitäten voll ausgenutzt werden.
Das Problem des maximalen Flusses in Netzwerken lässt sich effizient lösen. Da seine Struktur der Struktur unserer Aufgabe gleicht, lassen sich die Lösungsmethoden auch hier anwenden. So etwas kommt in der Informatik häufig vor: Ein Problem wird in ein anderes Problem mit der gleichen Struktur umgewandelt, in der man das Problem bereits einfach lösen konnte.

Stichwörter und Webseiten

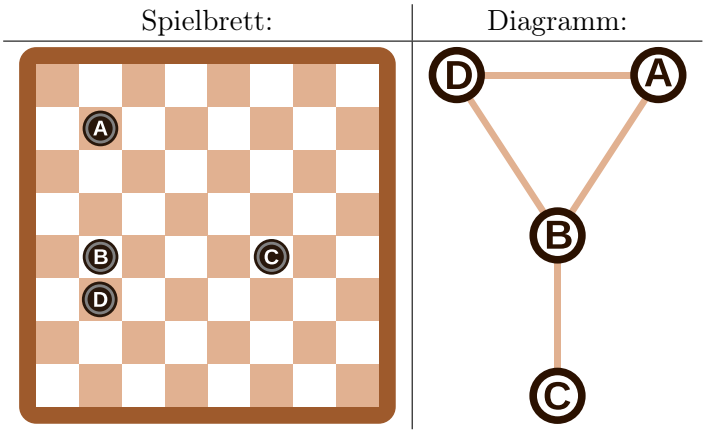
Netzwerkfluss, Problemreduktion

- `https://en.wikipedia.org/wiki/Maximum_flow_problem`

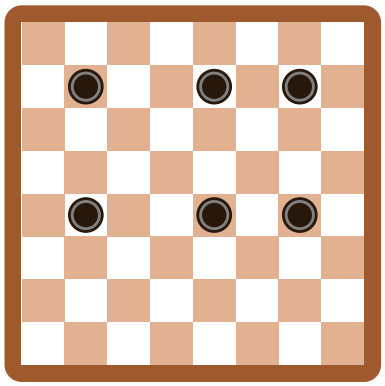
12. Zeilen und Spalten

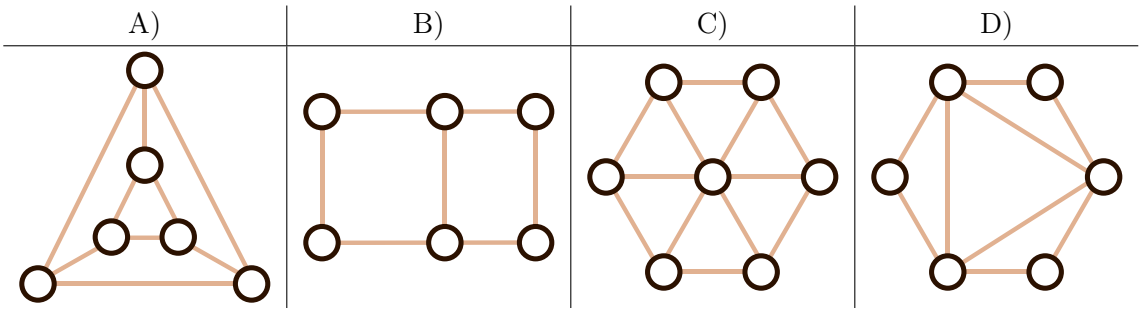
Aus den Spielsteinen auf dem Spielbrett wurde das Diagramm rechts vom Spielbrett so konstruiert, dass...

- ...jeder Spielstein durch einen Kreis dargestellt wird, und...
- ...2 Spielsteine im Diagramm durch eine Linie verbunden sind, wenn sie auf dem Brett in derselben Zeile oder in derselben Spalte liegen.



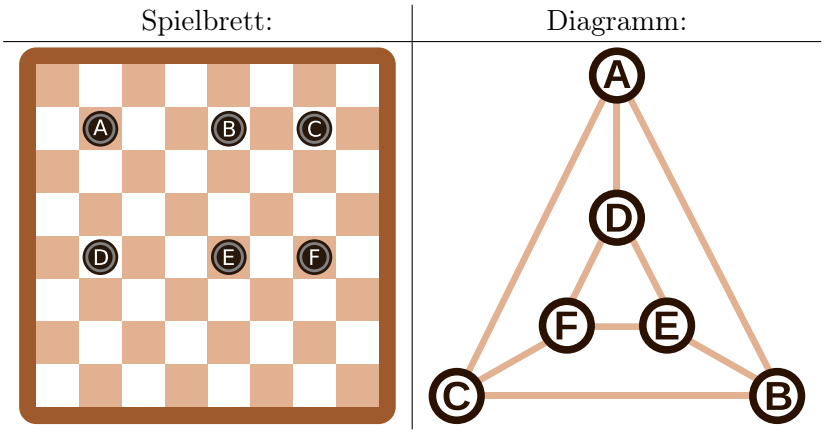
Die Spielsteine auf dem Spielbrett und die Kreise im Diagramm sind in diesem Beispiel mit Buchstaben bezeichnet, damit der Zusammenhang deutlich wird.
Welches Diagramm entspricht dem folgenden Spielbrett mit 6 Spielsteinen?



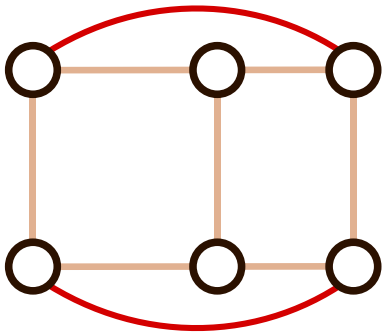


Lösung

Das Diagramm A) ist richtig. Das erkennt man in folgender Grafik, in der die Spielsteine mit Buchstaben markiert sind:



Man kann die Diagramme B), C) und D) mit folgender Beobachtung ausschliessen: Jeder Spielstein hat 2 andere Spielsteine in der selben Reihe und einen anderen Spielstein in der selben Spalte. Das heisst, dass jeder Spielstein im Diagramm mit genau $2 + 1 = 3$ anderen Spielsteinen verbunden sein muss. Das ist bei den anderen Diagrammen nicht der Fall. Zudem hat das Diagramm C) mit 7 Kreisen einen Kreis zu viel.
Das Diagramm B) ist nicht richtig, obwohl es dem Spielbrett ähnelt. Die äusseren 4 Kreise sind nur mit 2 anderen Kreisen verbunden. Um dieses Diagramm richtig zu machen, müsste man 2 zusätzliche Linien wie im folgenden Diagramm einzeichnen:



Dies ist Informatik!

In der Informatik werden solche Diagramme oft verwendet um das Wesentliche von Problemstellungen darzustellen. Solche Diagramme werden *Graphen* genannt. Die Kreise heissen *Knoten* und die Linien werden *Kanten* genannt.
Bei Graphen kommt es nur darauf an, welche Knoten mit welchen anderen Knoten durch Kanten verbunden sind. Die Anordnung der Knoten und auch die Form der Kanten spielt keine Rolle. Der selbe Graph kann daher auf unterschiedliche Weise dargestellt werden, wie wir bereits oben gesehen haben: Sowohl der Graph in Antwort A) als auch das letzte Bild in der Antworterklärung sind korrekte Lösungen und repräsentieren denselben Graph.
Graphen sind eine Form der Abstraktion. Sie repräsentieren das Wesentliche eines Problems. In unserem Fall kann man mit Hilfe des Graphen zum Beispiel das Problem „Was ist die kleinste Anzahl von Spielsteinen, die man entfernen muss, damit keine 2 Spielsteine in derselben Reihe oder derselben Spalte liegen?“ lösen. Eine wesentlicher Teil der Arbeit eines Informatikers liegt darin eine gute Repräsentation der Problemstellung zu finden, die zur Lösung des Problems hilfreich ist.

Stichwörter und Webseiten

Graph

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

13. Büchertausch

Jeder der drei Biber hat einen Tisch mit zwei Büchern. Sie wollen die Bücher durch Vertauschen benachbarter Bücher sortieren. Das machen die Biber gemeinsam in Runden. In jeder Runde darf ein Buch höchstens einmal bewegt werden.
Es gibt zwei verschiedene Typen von Runden, die immer abwechselnd durchgeführt werden:

A. Alle Biber dürfen (aber müssen nicht) die beiden Bücher auf seinem Tisch vertauschen (Beispiel A).

B. Die beiden linken Biber dürfen (aber müssen nicht) das rechte ihrer beiden Bücher mit linken Buch auf dem rechten Nachbartisch vertauschen (Beispiel B).

Die Biber beginnen mit folgender Anfangssituation:

Die erste Runde ist vom Typ A.
Wie viele Runden sind insgesamt mindestens notwendig um die Bücher zu sortieren, d.h. in die Reihenfolge 1, 2, 3, 4, 5, 6 *zu bringen?*

A) drei Runden

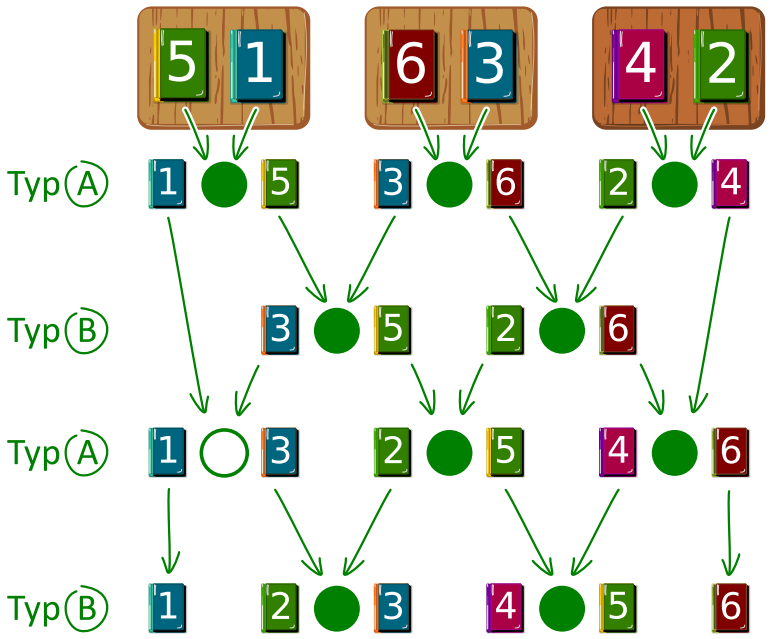
B) vier Runden

C) fünf Runden

D) sechs Runden

Lösung

Die richtige Antwort ist B).
Die Abbildung zeigt, wie die Bücher durch Platztausch sortiert werden können. Die Biber verfolgen eine „greedy-Strategie“ (“greedy” bedeutet „gierig“). Das heisst, sie versuchen bei jedem Schritt der Lösung ein Stückchen näher zu kommen. Sie vergleichen benachbarte Bücher, die gerade getauscht werden dürfen. Wenn diese beiden Bücher schon sortiert sind (das linke Buch hat eine kleinere Nummer als das rechte Buch), dann machen sie nichts. Ansonsten vertauschen sie die Bücher.
In der ersten Runde (Typ A) werden auf jedem Tisch die beiden Bücher vertauscht. In der zweiten Runde (Typ B) werden benachbarte Bücher von Nachbartischen getauscht, in der dritten Runde (Typ A) werden nur auf den beiden rechten Tischen die Bücher getauscht und in der vierten Runde (Typ B) werden alle benachbarten Bücher von Nachbartischen getauscht. Somit sind die Bücher sortiert. Schneller geht es nicht, denn das Buch 5 muss beispielsweise um vier Positionen in vier Runden nach rechts getauscht werden.



Dies ist Informatik!

Das Sortieren in dieser Aufgabe ist ein Beispiel für einen parallelen Algorithmus. Mehrere Akteure arbeiten zur gleichen Zeit an der Lösung eines Problems. Paralleles Sortieren kann durch ein Sortiernetz wie in der Abbildung dargestellt werden. Ein Sortiernetz besteht aus gerichteten Kanten, die durch Pfeile dargestellt werden, und Knoten, die durch die Kreise dargestellt werden.
In jeder Runde werden jeweils die beiden durch einen Kreis markierten Bücher verglichen und bei Bedarf vertauscht. Dabei können Vergleiche von Kreisen nebeneinander zeitgleich stattfinden. Wenn man den Pfeilen eines Buchs von oben nach unten folgt, kann man erkennen, wie es nach einigen Vertauschungen allmählich seine richtige Position in der angestrebten Reihenfolge einnimmt.

Stichwörter und Webseiten

Paralleles Sortieren, Sortiernetz

- `https://en.wikipedia.org/wiki/Sorting_network`

14. Soundex

Donald möchte Wörter nach ihrem Klang codieren. Er macht dazu Folgendes:

- Behalte den ersten Buchstaben bei.
- Streiche von allen anderen Buchstaben A, E, I, O, U, H, W und Y.
- Ersetze die restlichen Buchstaben wie folgt:
  - B, F, P oder V → 1
  - C, G, J, K, Q, S, X oder Z → 2
  - D oder T → 3
  - L → 4
  - M oder N → 5
  - R → 6
- Wenn nun zweimal oder öfters dieselbe Ziffer auftaucht, und die Buchstaben, die zu diesen Ziffern geführt haben, im Original direkt nebeneinander standen, behalte die Ziffer nur einmal. Dies gilt auch, wenn der erste Buchstabe durch diese Ziffer codiert würde, dann wird nur dieser Buchstabe behalten.
- Am Ende werden nur die ersten vier Zeichen (inkl. des ersten Buchstabens) notiert, fülle gegebenenfalls am Ende mit Nullen auf.

Die folgenden Wörter werden so codiert:

Euler → E460
Gauss → G200
Heilbronn → H416
Kant → K530
Lloyd → L300
Lissajous → L222

Welcher Code wird für das Wort „Hilbert“ erstellt?

A) H410

B) B540

C) H041

D) H416

Lösung

Der erste Buchstabe ist ein H, also ist das erste Zeichen des Codes ebenfalls ein H.
Danach werden alle A, E, I, O, U, H, W und Y gelöscht, nun muss also noch Hlbrt übersetzt werden.
Das Ersetzen der Buchstaben durch ihre Entsprechungen ergibt H4163.
Es gibt keine nebeneinanderliegenden Buchstaben mit demselben Code, also muss auch nichts gelöscht werden.
Nun werden die ersten vier Zeichen aufbewahrt, also ist H416 die richtige Antwort.

Dies ist Informatik!

Das Soundex-Verfahren, genauer gesagt der amerikanische Soundex, wurde bereits vor 100 Jahren von Robert C. Russel und Margaret King Odell entwickelt und patentiert. Es wurde dazu verwendet, ähnlich klingende Wörter in der englischen Sprache insbesondere auch ähnliche Namen von Personen zu finden. Das funktioniert, weil die Gruppen von Buchstaben, die demselben Code zugeordnet werden, ähnlich klingen: B, F, P und V sind Lippenlaute, C, G, J, K, Q, S, X und Z sind Gaumenlaute und Zischlaute, D und T sind Zahnlaute, L ist ein langer Fliesslaut, M und N sind Nasenlaute und R ist ein kurzer Fliesslaut.
Da es sehr einfach ist und nicht nur in der englischen Sprache relativ gute Resultate gibt, wird es häufig zur phonetischen Suche, also zur Suche nach ähnlich klingenden Wörtern verwendet. Es ist auch als Standard in vielen Datenbanken eingebaut.
Die Beispiele oben stammen von Donald Knuth, einem der ganz grossen Informatiker des 20. Jahrhunderts, der bis heute an seinem Buch „The Art of Computer Programming“ arbeitet. Im Band 3 „Sorting And Searching“ findet sich das beschriebene Verfahren.

Stichwörter und Webseiten

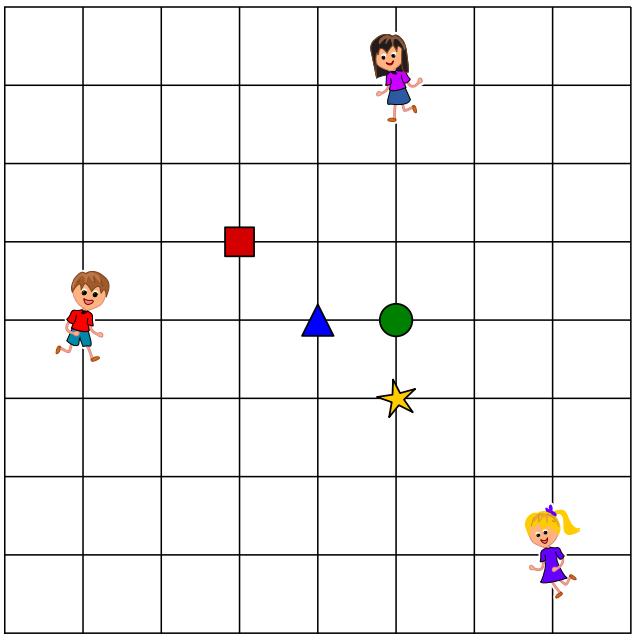
Phonetische Suche, Soundex

- `https://www.functions-online.com/soundex.html`
- `https://de.wikipedia.org/wiki/Soundex`
- `https://de.wikipedia.org/wiki/Laut`
- `https://www-cs-faculty.stanford.edu/~knuth/taocp.html`
- `http://www.highprogrammer.com/alan/numbers/soundex.html`

15. Drei Freunde

Alice  , Bob  und Céline  wohnen in La Chaux-de-Fonds. Sie haben auf dem Plan ihre Wohnorte markiert. Sie möchten einen Treffpunkt festlegen, zu dem die Summe ihrer Weglängen möglichst klein ist. Als Weglänge gilt die Zahl der Teilstrecken von Kreuzung zu Kreuzung.

 ,  ,  und  sind mögliche Treffpunkte. Der kürzeste Weg von Alice  zum Treffpunkt  hat beispielsweise die Länge 4.



Welches ist der Treffpunkt für den die Summe der Weglängen aller drei Freunde möglichst klein ist?

A)	B)	C)	D)
			

Lösung

Die richtige Antwort ist C)  . Die Summe der Weglängen der Freunde zum grünen Kreis beträgt: $3 + 4 + 5 = 12$.

Nicht richtig:  . Die Summe der Weglängen der Freunde zum roten Quadrat beträgt: $4+3+8 = 15$.

Nicht richtig:  . Die Summe der Weglängen der Freunde zum blauen Dreieck beträgt: $4+3+6 = 13$.

Nicht richtig:  . Die Summe der Weglängen der Freunde zum gelben Stern beträgt: $4+5+4 = 13$.

Dies ist Informatik!

Um von den vier Treffpunkten den besten zu bestimmen, muss für jeden von ihnen die Summe der Weglängen der drei Freunde berechnet werden. Der Treffpunkt mit der kleinsten Summe ist der beste Treffpunkt. Das ist einfach und kostet nicht viel Zeit – es geht sogar im Kopf. Bei einer grossen Anzahl von Freunden und einer grossen Zahl möglicher Treffpunkte wird daraus ein Optimierungsproblem, dessen Lösung in der Regel nicht in sinnvoller Zeit zu bestimmen ist.
In der Informatik gibt es Verfahren, die bei solchen Optimierungsproblemen in sinnvoller Zeit zumindest eine annähernd optimale Lösung finden, die beispielsweise nur noch maximal 1% von der optimalen Lösung entfernt ist. Wenn es anstelle vom Treffen von Freunden Zeitungsausträger sind, die an einem Ort ihre zu verteilenden Zeitungen abholen, kommt es bei 10 Minuten Wegzeit auf 6 Sekunden (1% von 10 Minuten) nicht an.
Ein solches Verfahren für eine annähernd optimale Lösung ist die lokale Suche: Um für eine grosse Zahl von Freunden einen annähernd optimalen Treffpunkt zu finden, wird bei der lokalen Suche zunächst irgend ein Vorschlag gesucht, der eventuell sogar zufällig gewählt wird. Davon ausgehend werden die Wegsummen für den ersten Vorschlag und für benachbarte Treffpunkte verglichen und der beste Treffpunkt in dieser Nachbarschaft bestimmt. So kann man sich dem Optimum annähern.
Warum wohnen die drei Freunde eigentlich in La Chaux-de-Fonds? La Chaux-de-Fonds ist zusammen mit Le Locle seit 2009 UNESCO Welterbe, und zwar nicht nur aufgrund der Geschichte der Uhrmacherei, sondern auch weil die Städte im 19. Jahrhundert nach Bränden ähnlich wie ein Schachbrett geplant und wieder aufgebaut wurden.

Stichwörter und Webseiten

Optimierungsproblem, lokale Suche

- `https://de.wikipedia.org/wiki/Lokale_Suche`
- `https://whc.unesco.org/en/list/1302`

A. Aufgabenautoren

Andrea Adamoli
Jared Asuncion
Daphne Blokhuis
Lucia Budinská
Špela Cerar
Kessarapan Charoensueksa
Kris Coolsaet
Valentina Dagienė
Christian Datzko
Susanne Datzko
Dilek Doğan
Marissa Engels
Hanspeter Erni
Gerald Futschek
Ionuț Gorgos
Shuchi Grover
Yasemin Gülbahar
Martin Guggisberg
Bent Halden
Urs Hauser
Juraj Hromkovič
Takeharu Ishizuka
Svetlana Jakšić
Zhang Jinbao
Dong Yoon Kim
Vaidotas Kinčius
Jia-Ling Koh
Regula Lacher
Anh Vinh Le
Dimitris Mavrovouniotis
Karolína Mayerová
Samart Moodleah
Tom Naughton
Henry Ong
Péter Piltmann
Zsuzsa Pluhár
Wolfgang Pohl
Ilya Posov
Nol Premasathian
J.P. Pretti
Doris Reck
Chris Roffey
Kirsten Schlüter
Andrea Maria Schmid
Jacqueline Staub
Allira Storey
Gabrielė Stupurienė
Peter Tomcsányi
Troy Vasiga
Rechilda Villame
Eslam Wageed
Pieter Waker
Michael Weigend
Magdalena Zarach

B. Sponsoring: Wettbewerb 2018

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/

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.

UBS

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

bbv Software Services

`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.

PRESENTEX
Das Geschenk – die gute Werbung

`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.

Bits with Bite
ZUBLER & PARTNER AG
Informatik

`http://www.zubler.ch/`
Zubler & Partner AG Informatik
Umfassendes Angebot an Dienstleistungen.

OXOCARD

`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.

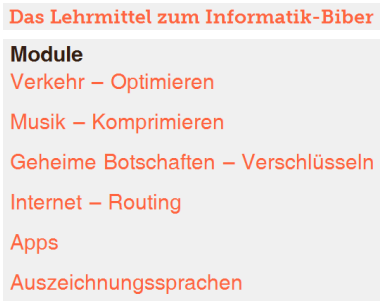
http://www.hepl.ch/
Haute école pédagogique du canton de Vaud

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



`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`

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.