

Paper Computer Science Experiment

9

<p>Great Principles of Computing</p> 	<h3>Automation (Suchalgorithmen)</h3>
 <p>Thema</p>	<h3>Labyrinth (Tiefensuche)</h3>
 <p>Unterrichtsform</p>	<p>Lernen am Modell</p>
 <p>Voraussetzung</p>	<p>In der antiken griechischen Sage muss der Athener Held Theseus den Minotaurus (ein Ungeheuer halb Mensch, halb Stier) töten. Zudem war der Minotaurus in einem Labyrinth versteckt. Die kluge Ariadne stattete ihren Geliebten mit einem langen Faden aus, den Theseus beim Durchforsten des Labyrinths abrollte. Damit konnte er vermeiden, gleiche Teile des Labyrinths mehrfach abzusuchen und zum anderen sicherstellen, auch wieder den Weg zurück in Ariadnes Arme zu finden.</p> 

	<p>Das begehbare Labyrinth auf dem Basler Leonhardskirchplatz Quelle: http://www.labyrinth-international.org/cms/index.php?page=1843268984&f=1&i=1843268984</p> <p>Nicht nur die alten Griechen mussten sich mit der Durchmusterung von Suchräumen befassen, sondern auch in der Informatik spielt diese Aufgabe eine zentrale Rolle (z.B. bei der Absuche und Indexierung des gesamten Internets durch Google). Eine Methode hierzu stellt die Tiefensuche (depth-first search) dar.</p>
 <p>Material</p>	<ul style="list-style-type: none"> • Labyrinth aus Styropor, Karton, Holz, Constri, Lego o.Ä. • Eine Spielfigur und Lottosteine oder andere Spielmarken.     <ul style="list-style-type: none"> • Kopiervorlage des Labyrinths auf Folie für die Demonstration der Froschperspektive.
 <p>Zeitdauer</p>	<p>Falls das Labyrinth bereits besteht: ca. 10 Min. Das Basteln eines eigenen Modells im Unterricht ist sehr zeitintensiv und darum nur während z.B. Projekttagen zu empfehlen.</p>
 <p>Vorgehen</p>	<p>Ariadne gab Theseus folgende Anweisung: Er solle in das Labyrinth hineinlaufen und dabei stets den Faden abwickeln, so dass er eine Spur seines Weges hinterlasse. An jeder Kreuzung solle er den am weitesten rechts liegenden Gang wählen. Geriete er in eine Sackgasse, solle er umkehren, den Faden wieder aufwickeln und an der nächsten Kreuzung die rechte Abzweigung nehmen, dort weitergehen, den Faden wieder abwickeln und so weiter.</p>



Der Ariadnefaden im Labyrinth

Quelle: Buch von Stephan Hußmann, Brigitte Lutz-Westphal: Kombinatorische Optimierung Erleben: In Studium und Unterricht, Vieweg, März 2007

Die Anweisungen Ariadnes können auf Papier oder, falls möglich, in einem echten oder (z. B. mit Tischen und Stühlen) selbst gebauten Labyrinth nachgespielt werden.

In den Skizzen und Modelle der eigenen Labyrinth und Wege wird auch einen Backtracking-Mechanismus sichtbar werden, nämlich dann, wenn der Faden wieder aufgewickelt wird.

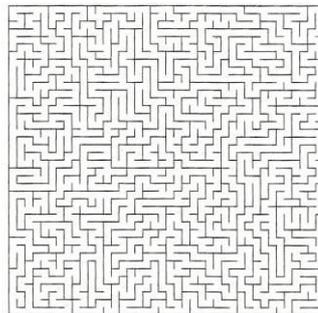
Die Tiefensuche kann zusätzlich zum Entwurf besonders verwirrender Labyrinth benutzt werden. Die Vorgehensweise ist recht einfach: Man startet mit einem regelmässigen rechteckigen Gitter. Die Tiefensuche startet in einer beliebigen Zelle



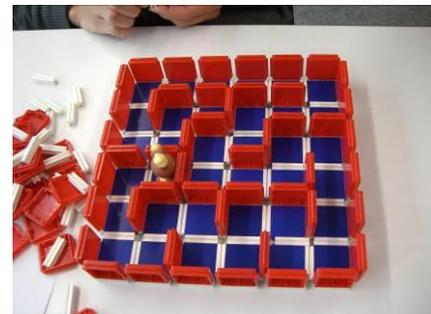
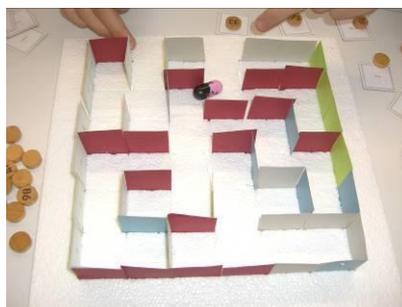
Dann wird die Tiefensuche für alle Nachbarzellen in zufälliger Reihenfolge aufgerufen. Wird dabei eine Zelle zum ersten Mal besucht, so wird die Wand zur Vorgängerzelle eingerissen.



Es ergeben sich Muster wie die folgenden:



Quelle: Buch von Berthold Vöcking, Helmut Alt, Martin Dietzfelbinger, Rüdiger Reischuk, Christian Schneideler, Heribert Vollmer, Dorothea Wagner: Taschenbuch der Algorithmen, Springer Verlag, 2008.



Da die Tiefensuche letztlich jede Zelle besucht, muss es von jeder Zelle einen Weg zur Startzelle geben und somit auch von jeder Zelle zu jeder anderen – nur leicht zu finden ist dieser Weg wie z.B. in der ersten Abbildung nicht unbedingt.



Varianten

Eine Schwierigkeit beim Niederschreiben von Algorithmen für die Tiefensuche liegt darin, dass man sehr viele Dinge intuitiv und durch „Draufgucken“ löst. Geht es um die Absuche von Labyrinthen, so ist das Hauptproblem, den Blick vom Gesamtbild zu lösen. Ein sehr wirksames Hilfsmittel dafür ist die Vorstellung der Froschperspektive welche sich im Schulhof oder in einem Maislabyrinth wunderbar erleben lässt.



2010 wurde in der jurassischen Hauptstadt Delémont das grösste Labyrinth der Schweiz eröffnet.

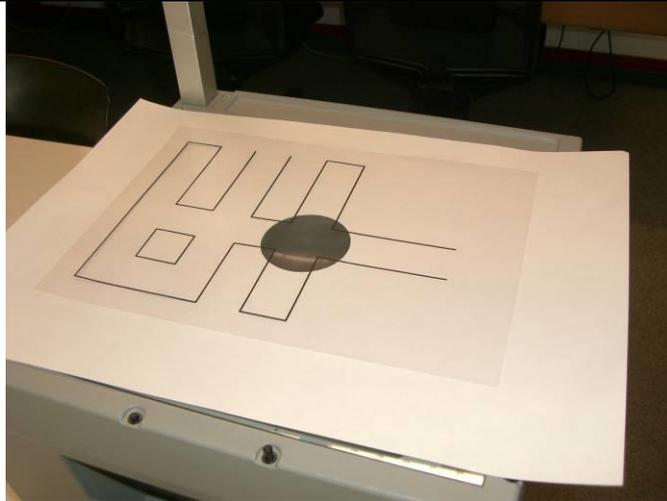
Quelle:

<http://www.schweizerfamilie.ch/unterwegs/weekendtipps/artikel/maislabirynth-die-suche-beginnt.html>

Es wird ein Labyrinth mit Kreide auf den Schulhof gezeichnet und eine Schülerin sucht den Weg durch dieses Labyrinth. Eine Gruppe von Mitschülern gibt dieser Person genaue Anweisungen, wie sie sich das Labyrinth bewegen soll. Der „Witz“ dabei ist, dass diese den Graphen auch nicht überblicken, bzw. im Idealfall gar nicht sehen. Die Person im Labyrinth braucht farbige Kreiden, um besuchte Kreuzungen und Wege markieren zu können. Alle anderen stehen, falls die räumlichen Gegebenheiten es zulassen, in einem höheren Stockwerk am Fenster und kontrollieren das Resultat.

Bei schlechtem Wetter hilft eine Lochblende weiter. Mit einem Labyrinth, der Lochblende und einem Stift oder bunter Kreide kann man sich ähnlich wie im Schulhof klarmachen, wie man den Computer trotz seiner Froschperspektive dazu bringt, den Weg durch das Labyrinth zu finden.

Ein Schüler steht an der Tafel oder dem OH-Projektor und die übrigen geben Anweisungen, was zu tun ist. Derjenige an der Tafel muss sich dabei ganz stur stellen, was sehr lustig sein kann. Die Rolle des „dummen Computers“ ist daher in vielen Klassen äusserst beliebt!



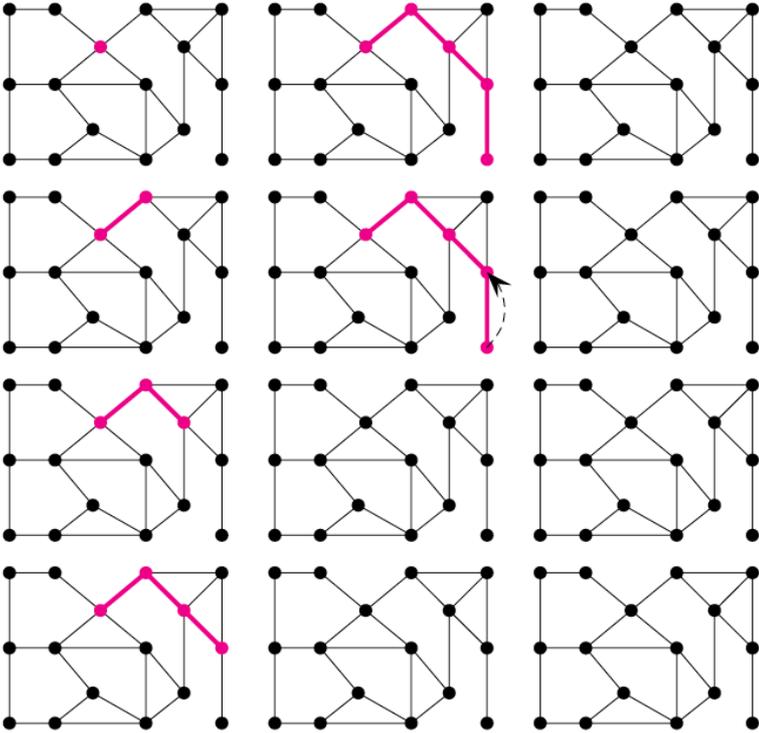
Die Lochblende kann beim Absuchen von Labyrinth verwendet werden

Die detaillierte Beschreibung des Algorithmus zur Absuche des Labyrinths auf dem obigen Bildes bzw. der Kopiervorlage findet sich im Taschenbuch der Algorithmen / Breiten- und Tiefensuche, S. 61 – 73

Das Nacherfinden der Tiefensuche ist der Einsatzbereich par excellence für das methodische Werkzeug „Lochblende“. Da diese Suche ganz linear verläuft, kann sie komplett mit der Lochblende durchgeführt werden.

Die Lochblende wird auf einen Graphen gelegt und die Tiefensuche ausgeführt, ohne die Lochblende hochzuheben. Wenn beim Backtracking gelöschte Kanten z.B. mit einer Wellenlinie durchgestrichen werden, so muss während der Prozedur nicht einmal den Stift angehoben werden. Er bleibt die ganze Zeit im Fenster der Lochblende. Der Verlauf des Algorithmus wird hier besonders anschaulich und (be-)greifbar.

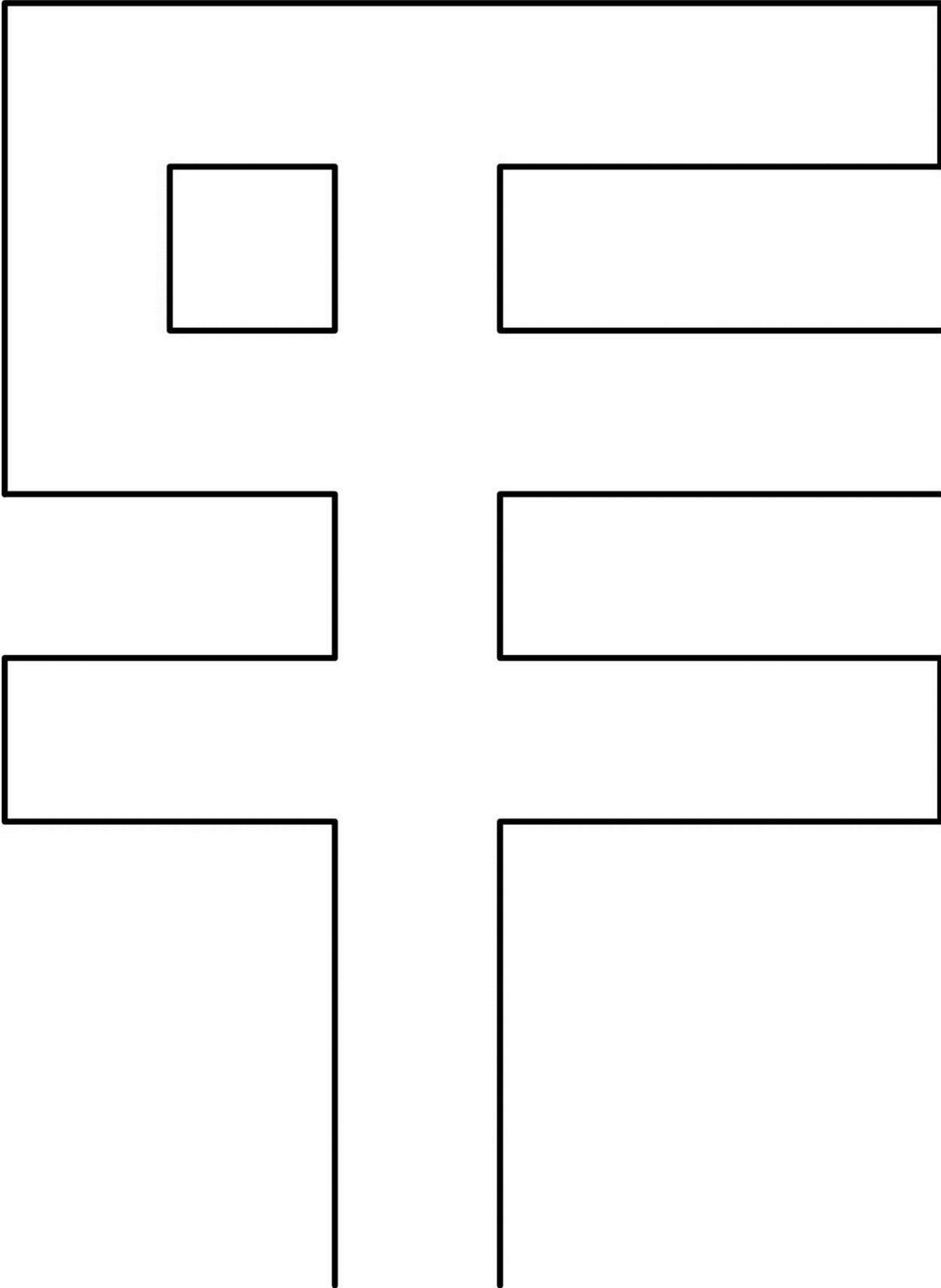
Ist ein Algorithmus gefunden, so lohnt es sich, zum noch tieferen Verständnis Daumenkinos basteln zu lassen. Dazu benötigt man reichlich Kopien ein und desselben Graphen und eine Klammermaschine. Aufgabe ist, pro Kopie des Graphen einen Schritt des Algorithmus einzuzichnen, so dass man sich den gesamten Ablauf des Algorithmus nachher als Daumenkino ansehen kann.

	 <p style="text-align: center;">Daumenkinos helfen, Algorithmen besser zu verstehen.</p> <p>Quelle: Buch von Stephan Hußmann, Brigitte Lutz-Westphal: Kombinatorische Optimierung Erleben: In Studium und Unterricht, Vieweg, März 2007</p> <p>Es wird auch deutlich, dass auf dem Graphen viele unterschiedliche aufspannende Bäume entstehen können, denn vermutlich wird die Tischnachbarin einen anderen Baum konstruieren als man selber. Aber alle aufspannenden Bäume sind gleichermaßen richtig. Diese Beobachtung führt direkt zur Frage nach der Anzahl der aufspannenden Bäume eines Graphen (Cayley-Formel: Die Anzahl der aufspannenden Bäume für den vollständigen Graphen mit n Knoten ist n^{n-2}).</p>
 <p>Weitere Ideen</p>	<p>Diverse im Unterricht anwendbare Beschreibungen von Algorithmen zur Tiefen- und Breitensuche finden sich hier: - Taschenbuch der Algorithmen Breiten-, Tiefensuche, S. 61 - 73</p> <p>Die Konstruktion eines Labyrinths mit dem oben beschriebenen Algorithmus kann auch sehr gut in Excel erzeugt werden:</p>

◇	A	B	C	D	E	F	G	H
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								Z
12								
13								
14								
15								
16								
17		S						
18								

Im Internet finden sich unzählige "Maze-Generators". Besonders zu empfehlen ist der folgende, welcher bei "Solve" auch den Backtracking Algorithmus zeigt:

<http://www.math.com/students/puzzles/mazegen/mazegen.html>



Quelle: Buch von Berthold Vöcking, Helmut Alt, Martin Dietzfelbinger, Rüdiger Reischuk, Christian Schneideler, Heribert Vollmer, Dorothea Wagner: Taschenbuch der Algorithmen, Springer Verlag, 2008.