Schwierige Probleme in der Informatik

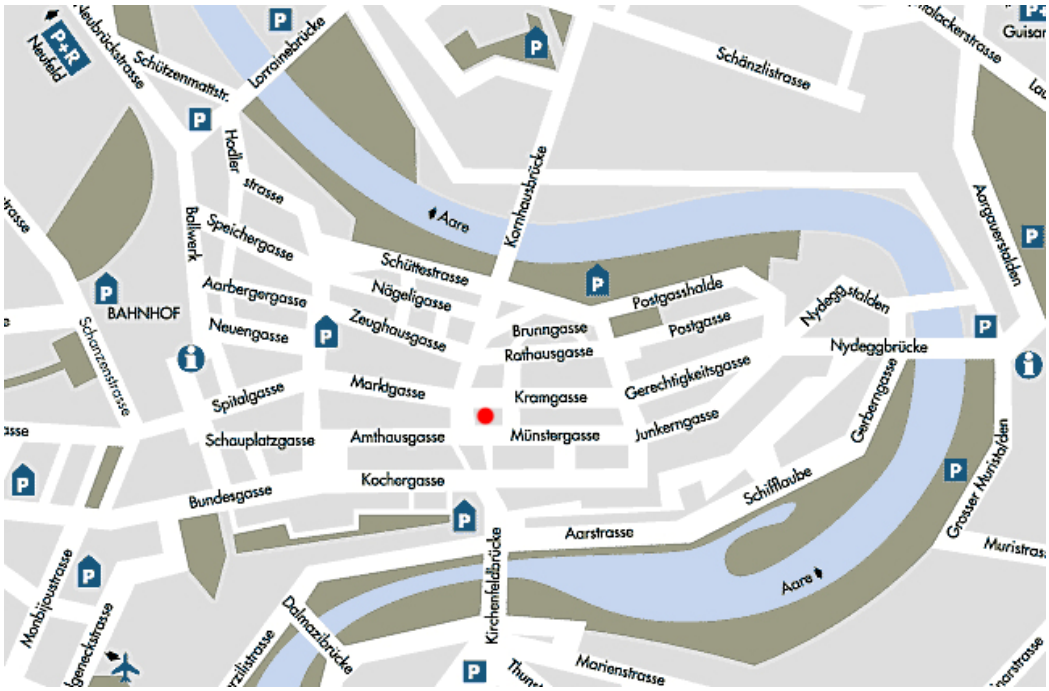
Ablauf

- Geschichte durchlesen 10 min
- Einführung in GraphBench 10 min
- Entdeckungsphase 60 min
- Diskussion mit Mitschüler
- Vorbereitung des Vortrags 30 min
- Kurzvorträge

Organisationsprobleme

Das letzte *Zürifäscht* war ein Riesenerfolg. Fast eine Million Leute strömten in die Metropole. Bern möchte, angetrieben durch diesen Erfolg, ein *Bärnfescht* organisieren.

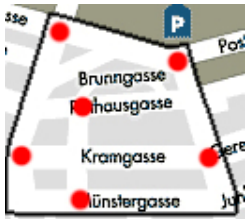
Der Stadtpräsident der Stadt Bern hat dafür ein Organisationskomitee einberufen, das Eure Klasse unterstützen soll.



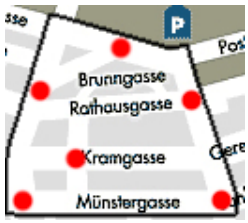
Berns Altstadt

Das *Bärnfescht* beginnt mit einem Kinderumzug durch Berns Altstadt. Der Umzug startet und endet beim *Zytglogge* (Punkt in der Mitte) und soll an jeder Kreuzung der Altstadt genau einmal vorbeiziehen. Die Route muss möglichst rasch bestimmt werden, um die Polizei zwecks Strassensperrung zu informieren.

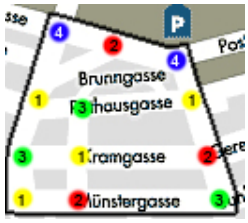
Nebst Attraktionen finden an diesem Fest viele Konzerte statt. Bühnen können aufgrund ihrer Grösse nur an Kreuzungen aufgestellt werden, sollen sich aber gegenseitig nicht beschallen. Das bedeutet, dass Bühnen nicht direkt durch eine Strasse verbunden sein dürfen. Die OK Mitglieder wissen nicht, wieviele Bühnen sie mieten müssen!



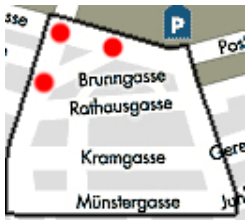
Die Polizei interessiert, wieviele Leute sie einsetzen muss. Die Polizisten werden auf Kreuzungen positioniert. Du denkst, dass es ausreicht, wenn jeder Strassenabschnitt von einem Polizisten überwacht wird. Durch Ausprobieren findest du die Lösung nach einer Stunde. Geht das effizienter?



Nebst Bühnen gibt es an den Kreuzungen auch je einen Stand (für Essens-, Getränke-, Vergnügungsstände und Toiletten). An benachbarten Kreuzungen sollen verschiedene Stände stehen. Ist das mit vier verschiedenen Ständen im Berner Strassennetz möglich?



Das Regionalfernsehen will das Fest live übertragen. Der Sender besitzt drei geeignete Kameras. Diese sind so auf Kreuzungen zu positionieren, dass jede Kamera mit den zwei anderen in Sichtkontakt steht, um die Strassen von beiden Seiten zu filmen. Welche Kreuzungen eignen sich dafür?



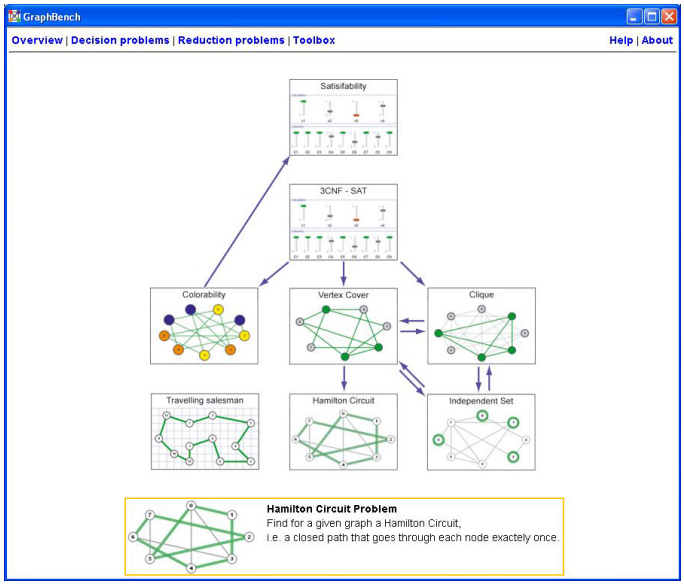
Ausserdem fliegt ein Werbezeppelin vor dem Fest über die wichtigsten Plätze und Sehenswürdigkeiten. Eine Runde soll so kurz wie möglich sein.

Du siehst, die Planung des *Bärnfeschts* wirft Probleme auf und ist eine spannende Angelegenheit.

All diese Fragen sind wichtige Informatikthemen und nicht einfach zu lösen. Man weiss zwar, wie man die Lösung bestimmt, je nach Grösse (Anzahl Knoten) kann es aber Tage dauern, bis die Lösung gefunden wird. Selbst mit dem schnellsten Computer!

Solche Probleme werden mit dem Programm GraphBench betrachtet. Starte das Programm und lies die folgende Einführung.

Einführung in GraphBench



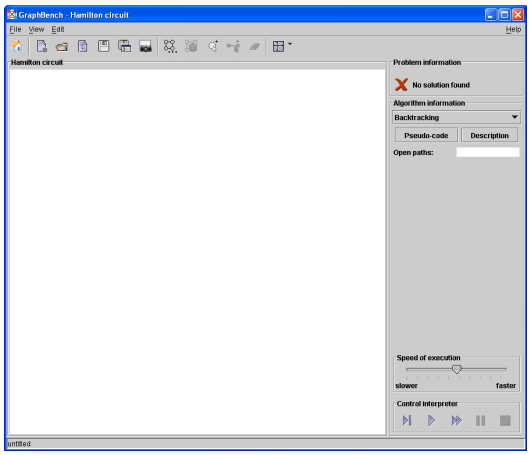
GraphBench ist eine Lernumgebung, die schwierige Probleme in der Informatik praktisch vermittelt. Die meisten Probleme lassen sich mit Graphen darstellen. Ein Graph besteht aus einer Menge von Knoten und Kanten. Eine Kante verbindet je zwei Knoten.

Das Programm präsentiert sich nach dem Start wie auf dem Bild links. Als erstes wählt man das zugrunde liegende Problem. Es sind hier nur die Entscheidungsprobleme (*Decision Problems*) von Bedeutung.

Folgende Probleme sind in GraphBench eingebaut:

- Colorability (Graphenfärbbarkeit)
- Vertex Cover (Knotenabdeckung)
- Clique (Komplette Subgraphen)
- Travelling Salesman (Problem des Handelsreisenden)
- Hamilton Circuit (Hamilton Kreis)
- Independent Set (Unabhängige Menge)
- Satisfiability (Erfüllbarkeit logischer Formeln)
- 3CNF-SAT (Satisfiability mit logischen Formeln in spezieller Form)

Für die Einführung in GraphBench klickst du auf das Bild mit dem Hamilton Circuit.



Du landest in der Ansicht des Hamilton Circuit Problems. Über die Symbolleiste kannst du Graphen erstellen, speichern und öffnen. Im rechten Teil werden die Algorithmen gesteuert und im Hauptfenster dargestellt.

Ohne Kenntnis der Tools kannst du mit GraphBench wenig lösen. In GraphBench ist eine englische Hilfe integriert. Du erreichst sie unter *Help -> GraphBench help*.

GraphBench Symbolbeschreibungen

 Zurück zum Startbildschirm

 Neue Datei erstellen

 Datei öffnen

 Datei neu laden

 Speichern

 Speichern als...

 Screenshot

Graphensymbole

 Grapheigenschaften ändern (Anzahl Knoten und Kanten)

 Zufälligen Graph erstellen (Anzahl Knoten bleibt gleich)

 Knoten hinzufügen

 Knoten mit Kante verbinden

 Knoten oder Kanten löschen

 Zoommöglichkeiten

 Sucht korrekte Färbung (Colorability Problem)

 Zufällige Knotenfärbung (Colorability Problem)

 Entfernt Farben aller Knoten (Colorability Problem)

Travelling Salesman Symbole

 Knoten hinzufügen

 Knoten löschen

 Knoten zufällig positionieren

 Zufällige Tour generieren

 Optimale Tour

 Kanten entfernen

 Tour entfernen

Satisfiability Symbole

 Formel randomisieren

 Eigenschaften der Formel ändern (Anzahl Variablen, Klauseln)

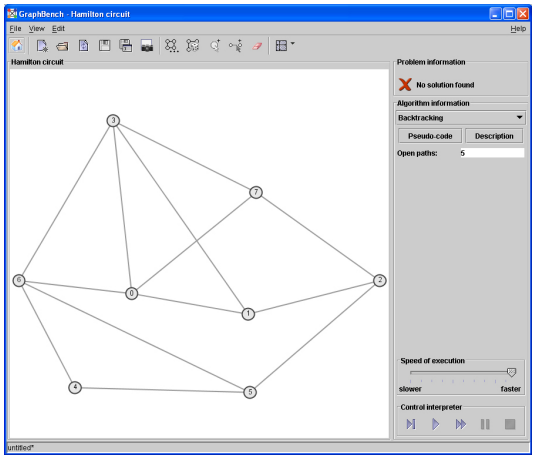
 Klausel hinzufügen

 Klausel löschen

 Variablen hinzufügen

 Variablen löschen

Mit den Symbolbeschreibungen bist du in der Lage, GraphBench zu nutzen. Versuche in den nächsten 5 Minuten den Graphen des untenstehenden Bildes zu erstellen.



Wähle anschliessend unter *Algorithm Information* einen Algorithmus aus und führe ihn aus (unter *Control interpreter*). Variiere die Geschwindigkeit (*Speed of execution*). Der *Pseudo-Code* Button liefert eine Beschreibung des Algorithmus' in einer abstrahierten Programmiersprache, der *Description* Button eine kurze theoretische Beschreibung in Englisch.

Gehe zurück zum Startbildschirm und verschaffe dir einen Überblick über die vorhandenen Entscheidungsprobleme. Wähle eines aus, mit dem du dich in den folgenden 60 Minuten beschäftigen wirst. Die einzelnen Umgebungen unterscheiden sich leicht voneinander.

Deine Aufgabe

Mach dich mit dem Problem vertraut: Spiel mit der Umgebung und finde heraus, worum es hier geht. Du sollst die Lösungsidee einem Nicht-Informatiker erklären können. Hinweis: Auf der letzten Seite ist jedes Problem kurz beschrieben.

Überlege dir Fragen zu diesem Problem und versuche sie zu beantworten. Eine Auswahl von Fragen ist aufgeführt. Es gibt viele andere Gesichtspunkte, die du untersuchen kannst:

- Wie funktionieren die Algorithmen?
- Kann ich selbst einen Algorithmus finden, der das Problem löst?
- Wieviele Schritte braucht der Algorithmus, bis er eine Lösung findet?
- Welcher Algorithmus ist der schnellste (d.h. benötigt am wenigsten Schritte)?
- Findet das Programm immer die richtige Lösung?
- Wie siehts aus mit Extremfällen?
- Gibt es praktische Anwendungen für dieses Problem?
- Gibt es Grenzen in der praktischen Nutzung?

Formuliere deine Antworten und Erkenntnisse in kurzen, einfachen Sätzen. Eine Person, die sich nicht mit Informatik beschäftigt, sollte deine Erkenntnisse verstehen.

Nach 60 Minuten setzt du dich mit einem Mitschüler zusammen, der dasselbe Problem studiert hat. Ihr diskutiert, was ihr entdeckt habt.

Anschliessend überarbeitest du deine Erkenntnisse. Formuliere 3-5 Erkenntnisse und halte sie schriftlich fest.

Bereite einen 3-minütigen Kurzvortrag zu deinem Thema vor. Er beinhaltet die Schilderung des Problems, einen möglichen Lösungsalgorithmus und deine Erkenntnisse. Der Vortrag muss von deinen Mitschülern verstanden werden. Dafür hast du ca. 30 Minuten Zeit.

Viel Spass beim Entdecken!

Beschreibung der Problemstellungen

- **Travelling Salesman Problem**
  Eine Tour besucht jeden Punkt eines Graphen. Die Länge der Verbindung zweier Punkte zeigt deren Entfernung. Finde für einen gegebenen Graphen die kürzeste Tour.
- **Hamilton Circuit (Hamilton Kreis)**
  Ein Hamilton Kreis ist ein Pfad, der jeden Knoten genau einmal besucht und jede Kante höchstens einmal benutzt. Die Frage ist, ob in einem Graphen ein Hamilton Kreis existiert.
- **Colorability (Graphenfärbbarkeit)**
  In einem Graphen werden die Knoten eingefärbt. Keine benachbarten Knoten (zwei durch Kanten verbundene Knoten) dürfen dieselbe Farbe aufweisen. Dieses Problem beantwortet die Frage ob es eine Färbung mit *n* Farben für einen bestimmen Graphen gibt. Auch die Frage nach der minimalen Anzahl Farben lässt sich beantworten.
- **Clique**
  Eine Clique ist eine Menge von Knoten, in der jeder Knoten mit allen anderen Knoten verbunden ist. Eine solche Knotenmenge wird "vollständiger Graph" genannt. Es soll eine Clique mit einer vorgegebenen Anzahl Knoten in einem Graphen gefunden werden.
- **Independent Set (Unabhängige Menge)**
  Knoten sind unabhängig, wenn sie nicht benachbart, d.h. nicht direkt durch eine Kante verbunden sind. In einer unabhängigen Menge von Knoten ist kein Knoten abhängig von einem anderen. Finde eine vorgegebene Menge unabhängiger Knoten. Wie gross ist das maximale Independent Set?
- **Vertex Cover (Knotenabdeckung)**
  Eine Untermenge von Knoten soll alle Kanten abdecken. Eine Kante ist abgedeckt, falls der Anfangs- oder Endknoten der Kante ausgewählt ist. Mit wievielen Knoten ist eine vollständige Abdeckung möglich? Geht es mit 3 Knoten? Um dieses Szenario geht es im Vertex-Cover Problem.
- **Satisfiability (Erfüllbarkeit logischer Formeln)**
  Den Variablen in einer logischen Formel werden Wahr/Falsch Werte so zugeordnet, dass die ganze Formel erfüllt ist.
- **3-CNF Satisfiability**
  Wie Satisfiability, nur dass die Formel in einer speziellen Form (3-CNF) gegeben ist.
  Eine Formel ist in CNF, wenn man sie als Konjunktion von Disjunktionen schreiben kann. Eine logische Formel in 3-CNF enthält zusätzlich höchstens 3 Literale pro Klausel.