Die Eine-Million-Dollar Frage
Ist P = NP?

Kurt Mehlhorn

max planck institut
informatik

SIC Saarland
Informatics Campus

Wir wissen alle:

Das Finden einer Lösung für eine Aufgabe ist schwerer als das Überprüfen eines Lösungsvorschlags.

Finden versus Überprüfen einer Lösung

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

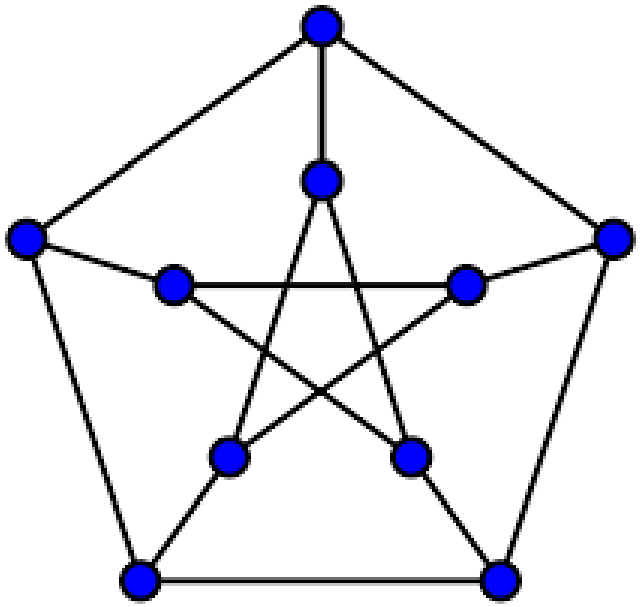
Ein Sudoku: Jedes Quadrat soll mit einer Zahl zwischen 1 und 9 beschriftet werden, so dass in jeder Zeile, in jeder Spalte und in jedem Subquadrat jede Zahl genau einmal vorkommt.

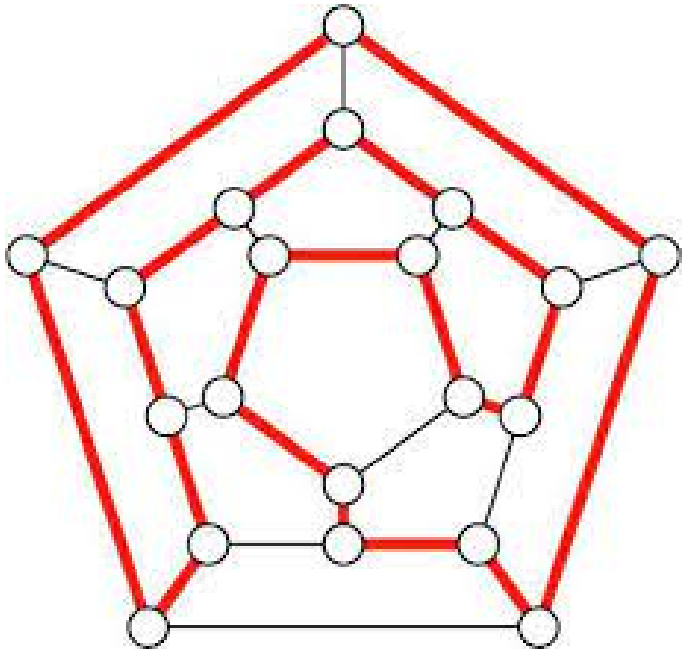
Finden versus Überprüfen einer Lösung

Ein Sudoku: Jedes Quadrat soll mit einer Zahl zwischen 1 und 9 beschriftet werden, so dass in jeder Zeile, in jeder Spalte und in jedem Subquadrat jede Zahl genau einmal vorkommt.

Das Hamiltonsche Kreis Problem

Gegeben ein Graph, gib eine Tour an, die jeden Knoten genau einmal besucht.





Überprüfen eines Lösungsvorschlags ist einfach, aber Finden einer Lösung scheint sehr schwer.

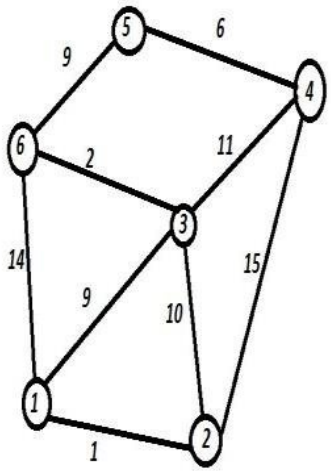
P = NP?

- Effiziente und ineffiziente Algorithmen
- Die Welt vor 1971
- Cook, Levin und Karp bringen Struktur ins Chaos: NP-Vollständigkeit
- Die Eine-Million-Dollar Frage: Ist P = NP?
- Was wäre, wenn P $\neq$ NP?
- Was wäre, wenn P $=$ NP?
- Wie umgehen mit NP-Vollständigkeit?

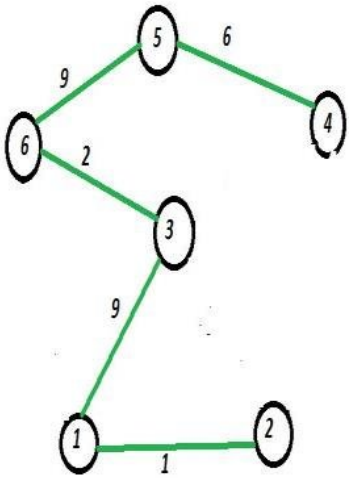
Die Welt vor 1971

Man kannte für einige Probleme effiziente Algorithmen

- Multiplikation von ganzen Zahlen
- Finden des kürzesten Weges von A nach B.
- Minimaler Spannbaum
- **P** = alle Probleme, für die es einen effizienten Alg gibt.



Undirected graph G = (V,E)



The minimal spanning tree
The tree cost is 33

Die Welt vor 1971

Man kannte für einige Probleme effiziente Algorithmen

- Multiplikation von ganzen Zahlen
- Finden des kürzesten Weges von A nach B.
- Minimaler Spannbaum
- **P** = alle Probleme, für die es einen effizienten Alg gibt.

Effizienter Algorithmus: Man kann sehr große Problemstellungen in wenigen Sekunden lösen.

- Zahlen mit einer Million Stellen, kürzeste Wege im Straßengraph von Europa (5 Millionen Knoten), Spannbaum des Straßengraphen von Europa.
- Wenn man die Problemgröße verdoppelt, erhöht sich die Laufzeit um einen konstanten Faktor (2, 4, . . .).
- Formal: Laufzeit ist beschränkt durch ein Polynom in der Größe der Eingabe.

Für viele Probleme kannte man nur ineffiziente Algorithmen

Beispiele

- Hamiltonsches Kreis Problem
- Problem des Handlungsreisenden: Gegeben eine Menge von Städten, gib eine Rundreise an, die eine vorgegebene Länge nicht übersteigt.

- Rucksackproblem
- Erfüllbarkeitsproblem der Aussagenlogik: kommt später.
- Allgemein: Probleme, für die man Lösungskandidaten effizient überprüfen kann.

Für viele Probleme kannte man nur ineffiziente Algorithmen

Beispiele

- Hamiltonsches Kreis Problem
- Problem des Handlungsreisenden
- Rucksackproblem: Gegeben Objekte, jeweils mit Gewicht und Wert, und eine Gewichtsgrenze, gib eine Teilmenge der Objekte an, die einen vorgegebenen Gesamtwert hat.

$$(g, w) = (3, 2), (4, 1), (2, 4), \quad G \leq 5, \quad W \geq 5$$

- Erfüllbarkeitsproblem der Aussagenlogik: kommt später.
- Allgemein: Probleme, für die man Lösungskandidaten effizient überprüfen kann.

Für viele Probleme kannte man nur ineffiziente Algorithmen

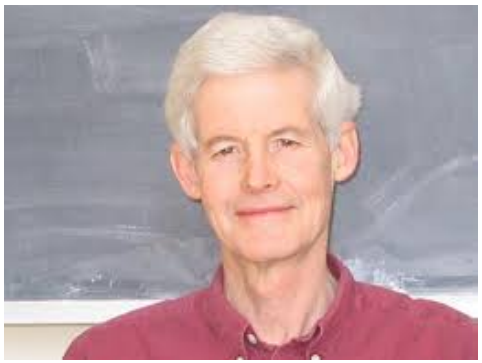
Beispiele

- Hamiltonsches Kreis Problem
- Problem des Handlungsreisenden
- Rucksackproblem
- Erfüllbarkeitsproblem der Aussagenlogik: kommt später.
- Allgemein: Probleme, für die man Lösungskandidaten effizient überprüfen kann.

Sehr ineffizienter Algorithmus

- Nur kleine Probleme sind lösbar, Problemgröße 1000.
- Wenn man die Problemgröße um eins erhöht, verdoppelt sich die Laufzeit.
- Man scheint alle Lösungskandidaten durchprobieren zu müssen.

Steven Cook, Leonid Levin und Richard Karp brachten Ordnung ins Chaos



Steven Cook
Turing Award

Leonid Levin
Knuth Prize

Richard Karp
Turing Award

Ihre Ergebnisse wurden innerhalb von 2 Jahren Allgemeingut.

Einzug in Lehrbücher und Grundvorlesungen.

Der Satz von Cook/Levin, 1971

Satz (Stephen Cook und Leonid Levin, 1971)

Wenn es einen effizienten Algorithmus für das Erfüllbarkeitsproblem der Aussagenlogik gibt, dann gibt es auch einen solchen für den Handlungsreisenden, für Hamiltonschen Kreis, für Graphenfärben,

*für alle Probleme in **NP**.*

Man sagt: Das Erfüllbarkeitsproblem ist NP-vollständig.

Die Klasse **NP**

Ein Problem ist in **NP**, wenn man für jeden Lösungskandidaten effizient überprüfen kann, ob er tatsächlich eine Lösung ist.

Theorie schafft Einsicht. Theorie ordnet.

Das Erfüllbarkeitsproblem SAT (Logelei von Zweistein)

Ruth erzählt ihren drei Freundinnen, dass sie es immer schwer findet, aus dem, was ihr Mann sagt, schlau zu werden. . . .

Pia: Sid hat keinen Amiga, Vic macht nichts mit Hardware, Pal hat keinen Z80, Sid ist kein Gamer, und Vic hat keinen C64.

Mia: Vic macht nichts mit Hardware, Sid hat keinen Z80, Pal macht nichts mit Hardware, Vic hat keinen Amiga, und Sid programmiert.

Ria: Pal hat einen C64, Sid hat keinen Amiga, Sid ist kein Gamer, Vic hat keinen C64, und Pal programmiert.

Von den fünf Aussagen sind immer nur genau zwei richtig! Wie lautet die Zuordnung?

	Z80	C64	Amiga	Gamer	Programmierer	Hardware
Sid						
Vic						
Pal						

Das Erfüllbarkeitsproblem SAT (Logelei von Zweistein)

Ruth erzählt ihren drei Freundinnen, dass sie es immer schwer findet, aus dem, was ihr Mann sagt, schlau zu werden. . . .

Pia: Sid hat keinen Amiga, Vic macht nichts mit Hardware, Pal hat keinen Z80, Sid ist kein Gamer, und Vic hat keinen C64.

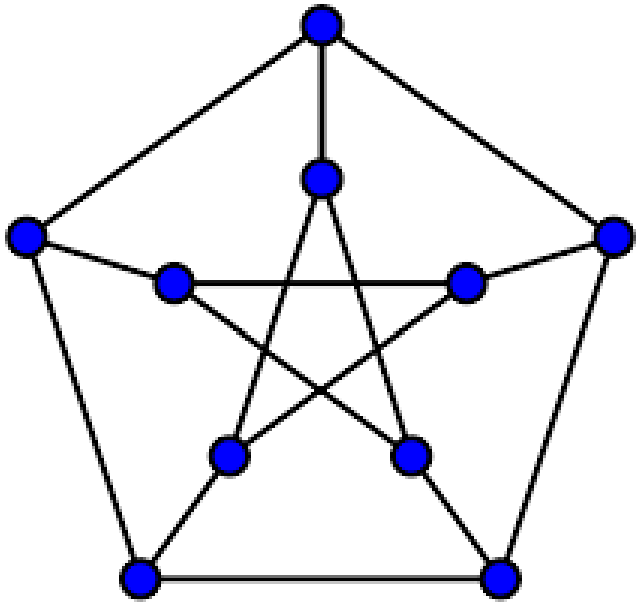
Mia: Vic macht nichts mit Hardware, Sid hat keinen Z80, Pal macht nichts mit Hardware, Vic hat keinen Amiga, und Sid programmiert.

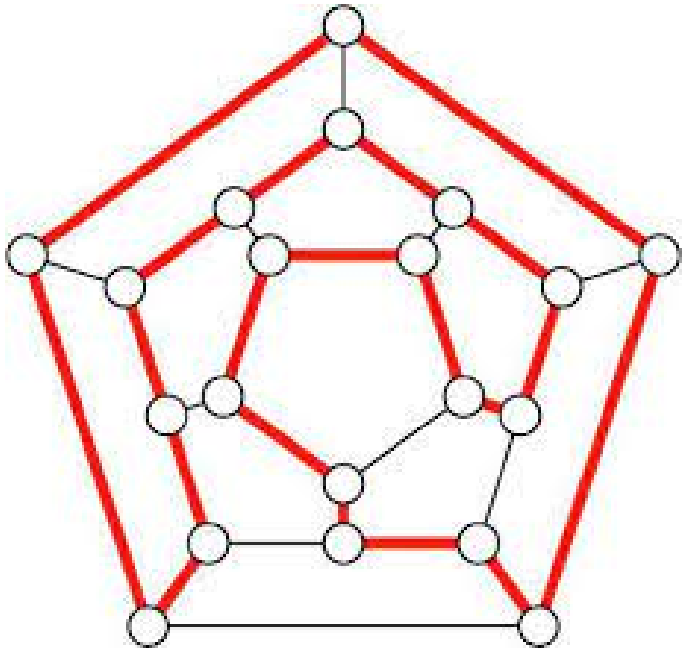
Ria: Pal hat einen C64, Sid hat keinen Amiga, Sid ist kein Gamer, Vic hat keinen C64, und Pal programmiert.

Von den fünf Aussagen sind immer nur genau zwei richtig! Wie lautet die Zuordnung?

	Z80	C64	Amiga	Gamer	Programmierer	Hardware
Sid	X			X		
Vic		X				X
Pal			X		X	

Hamiltonscher Kreis $\leq$ Erfüllbarkeitsproblem



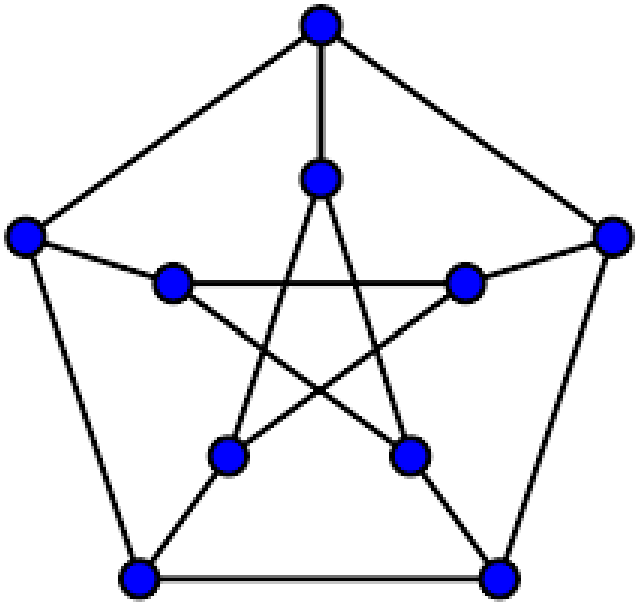


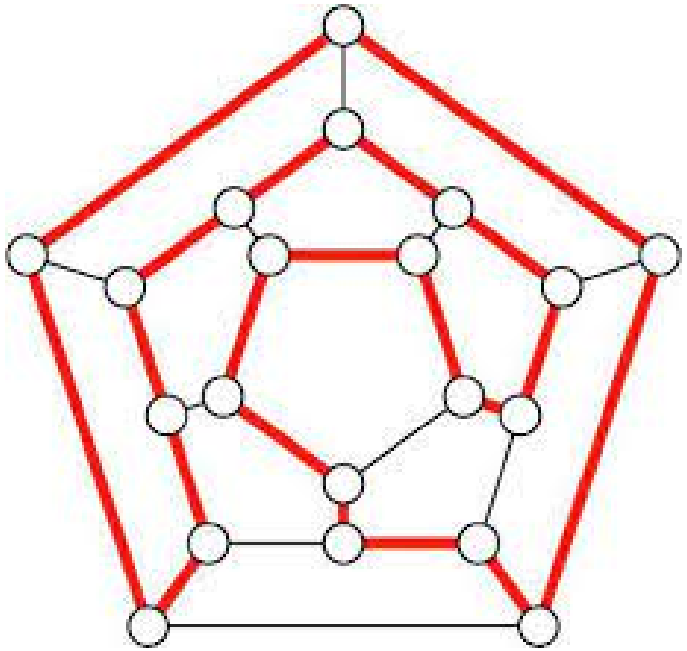
Das Hamiltonsche-Kreis-Problem ist eine Logelei.

Wähle eine Menge von Kanten, so dass

- für jeden Knoten genau zwei anliegende Kanten und
- jeder Knoten über ausgewählte Kanten vom Knoten 1 aus erreichbar.

Hamiltonscher Kreis $\leq$ Erfüllbarkeitsproblem





Das Hamiltonsche-Kreis-Problem ist eine Logelei.

- Knoten 1 ist in Null Schritten erreichbar und keiner sonst.
- Knoten i ist in k Schrittten erreichbar, wenn er mit einem Knoten benachbart ist, der schon in $k-1$ Schritten erreichbar ist.

NP-Vollständigkeit (Satz von Karp)

Die Klasse **NP**

Ein Problem ist in **NP**, wenn man für jeden Lösungskandidaten effizient überprüfen kann, ob er tatsächlich eine Lösung ist.

Cook-Levin: Das Erfüllbarkeitsproblem ist NP-vollständig, d.h., wenn es einen effizienten Algorithmus für SAT gibt, dann für alle Probleme in **NP**.

Satz (Karp, 1972)

Das Graphenfärbungsproblem, das Hamiltonsche Kreisproblem, Knapsack und 20 andere Probleme sind NP-vollständig.

Die Liste ist inzwischen auf mehrere Tausend angewachsen.

Theorie schafft Ordnung.

Zwischenstand

Das **P** = **NP** Problem: Gibt es einen effizienten Algorithmus für das Erfüllbarkeitsproblem?

Die Clay Foundation hat 1 Million Dollar ausgelobt für die Lösung als eines der 7 größten offenen Probleme der Mathematik.

Angebliche Lösungen erscheinen ständig:

- monatlich: Gleichheit durch Angabe eines (angeblich) effizienten Algorithmus für ein NP-vollständiges Problem.
- halbjährlich: Ungleichheit durch Ableiten eines Widerspruchs aus Gleichheit.

Was wäre,

- wenn **P = NP** wäre,
- wenn **P** $\neq$ **NP** wäre?

Eines davon ist richtig. Wir wissen nur nicht, welches von beiden.

Bessere Frage: Was wäre, wenn man

- einen Beweis dafür hätte, dass **P** $\neq$ **NP** ist.
- man einen effizienten Algorithmus für das Erfüllbarkeitsproblem hätte.

Was wäre,

- wenn **P = NP** wäre,
- wenn **P** $\neq$ **NP** wäre?

Eines davon ist richtig. Wir wissen nur nicht, welches von beiden.

Bessere Frage: Was wäre, wenn man

- einen Beweis dafür hätte, dass **P** $\neq$ **NP** ist.
- man einen effizienten Algorithmus für das Erfüllbarkeitsproblem hätte.

Was wäre, wenn man einen Beweis für P $\neq$ NP hätte?

- Es würde sich nicht viel ändern.
- Da wir keinen effizienten Algorithmus für das Erfüllbarkeitsproblem kennen, leben wir faktisch in einer Welt, in der P ungleich NP ist.
- Die meisten Fachleute glauben, dass P $\neq$ NP: Lösen ist schwerer als Überprüfen.
- Aber: Im Augenblick gibt es keinen Ansatz, wie man P $\neq$ NP beweisen könnte. Man weiß nur, dass einige natürliche Ansätze NICHT funktionieren können.

  Wenn man einen Beweis findet, muss dieser eine neue Methode einführen. Diese Methode könnte weitere Anwendungen haben.
- Jedes halbe Jahr wird ein (falscher) Beweis angekündigt.

Was wäre, wenn man einen effizienten Algorithmus für SAT hätte?

- Das wäre eine Revolution.
- Wir hätten effiziente Algorithmen für Erfüllbarkeit, ...

- Mathematiker würden arbeitslos:

  Input: Ein mathematischer Satz S, eine Anzahl n unbeschriebener Blätter

  Frage: Gibt es einen Beweis für S (in einem formalen System), der auf die n Blätter passt?

  Formales System = Korrektheit eines Beweises effizient entscheidbar

  Dieses Problem ist in NP. Falls P = NP, dann ist dieses Problem in P.

Was wäre, wenn man einen effizienten Algorithmus für SAT hätte?

- Das wäre eine Revolution.
- Wir hätten effiziente Algorithmen für Erfüllbarkeit, ...

- Kryptographie würde nicht mehr funktionieren.

$$\text{Klartext} \rightarrow \text{Verschlüsselung} \rightarrow \text{verschlüsselter Text}$$

Frage: Was ist der Klartext zu einem verschlüsseltem Text?

Diese Problem ist in **NP**. Also gäbe es einen effizienten Algorithmus dafür.

Was wäre, wenn man einen effizienten Algorithmus für SAT hätte?

- Das wäre eine Revolution.
- Wir hätten effiziente Algorithmen für Erfüllbarkeit, ...
- Philosophen müssten neu über den Begriff Kreativität nachdenken. Beweisen versus Prüfen.

Wie geht man mit NP-Vollständigkeit um?

Für viele praktisch wichtige Probleme kennen wir keinen effizienten Algorithmus und werden meiner Meinung nach auch nie einen finden. Aber, nur weil ein Problem schwer ist, verschwindet es nicht.

Ein Grund für Depression?

Nein, etwas geht immer. Jetzt erst recht.

Man muss NP als eine Herausforderung sehen.

NP-Vollständigkeit bedeutet: Man kennt keinen Algorithmus, der jede Problemstellung in Polynomzeit löst. Es kann durchaus Algorithmen geben, die viele (interessante) Instanzen effizient lösen oder zumindest sehr gute Lösungen finden.

Wie geht man mit NP-Vollständigkeit um?

Für viele praktisch wichtige Probleme kennen wir keinen effizienten Algorithmus und werden meiner Meinung nach auch nie einen finden. Aber, nur weil ein Problem schwer ist, verschwindet es nicht.

Ein Grund für Depression?

Nein, etwas geht immer. Jetzt erst recht.

Man muss NP als eine Herausforderung sehen.

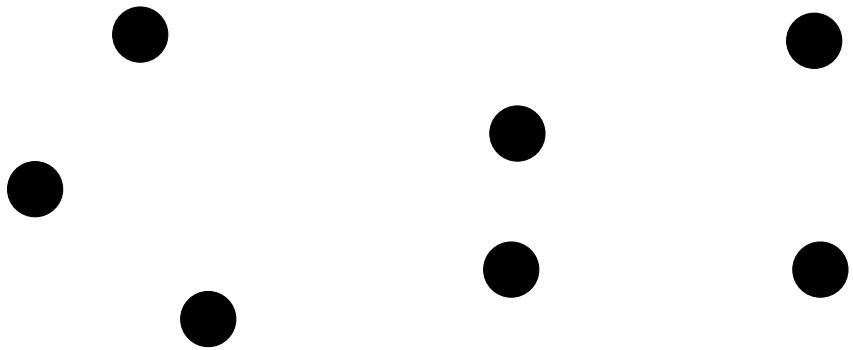
NP-Vollständigkeit bedeutet: Man kennt keinen Algorithmus, der jede Problemstellung in Polynomzeit löst. Es kann durchaus Algorithmen geben, die viele (interessante) Instanzen effizient lösen oder zumindest sehr gute Lösungen finden.

Jetzt erst recht am Beispiel des Problems des Handlungsreisenden

Gegeben Punkte in der Ebene. Man finde eine kürzeste Rundreise durch alle Punkte.

Eine Heuristik:

- Starte irgendwo und gehe immer zur nächstgelegenen noch nicht besuchten Stadt.
- Wenn alle besucht, kehre zum Ausgang zurück.
- Entkreuze.



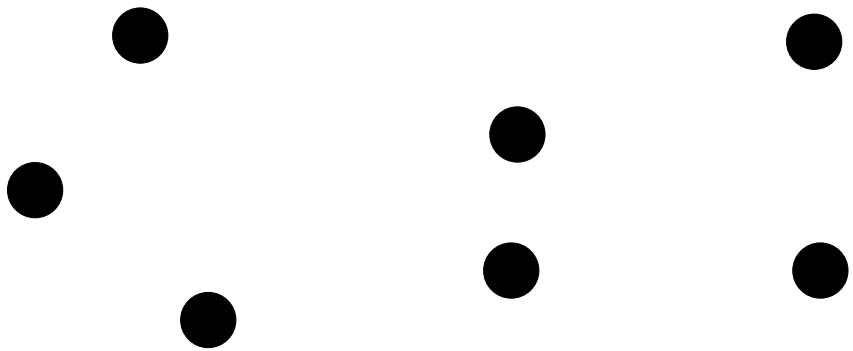
Es gibt viel bessere Heuristiken, z.B. Lin-Kernighan.

Jetzt erst recht am Beispiel des Problems des Handlungsreisenden

Gegeben Punkte in der Ebene. Man finde eine kürzeste Rundreise durch alle Punkte.

Approximationsalgorithmen:

- Konstruiere einen minimalen aufspannenden Baum.
- Lauf einmal außen um den Baum herum.
- Höchstens zweimal so lange, wie eine optimale Tour.

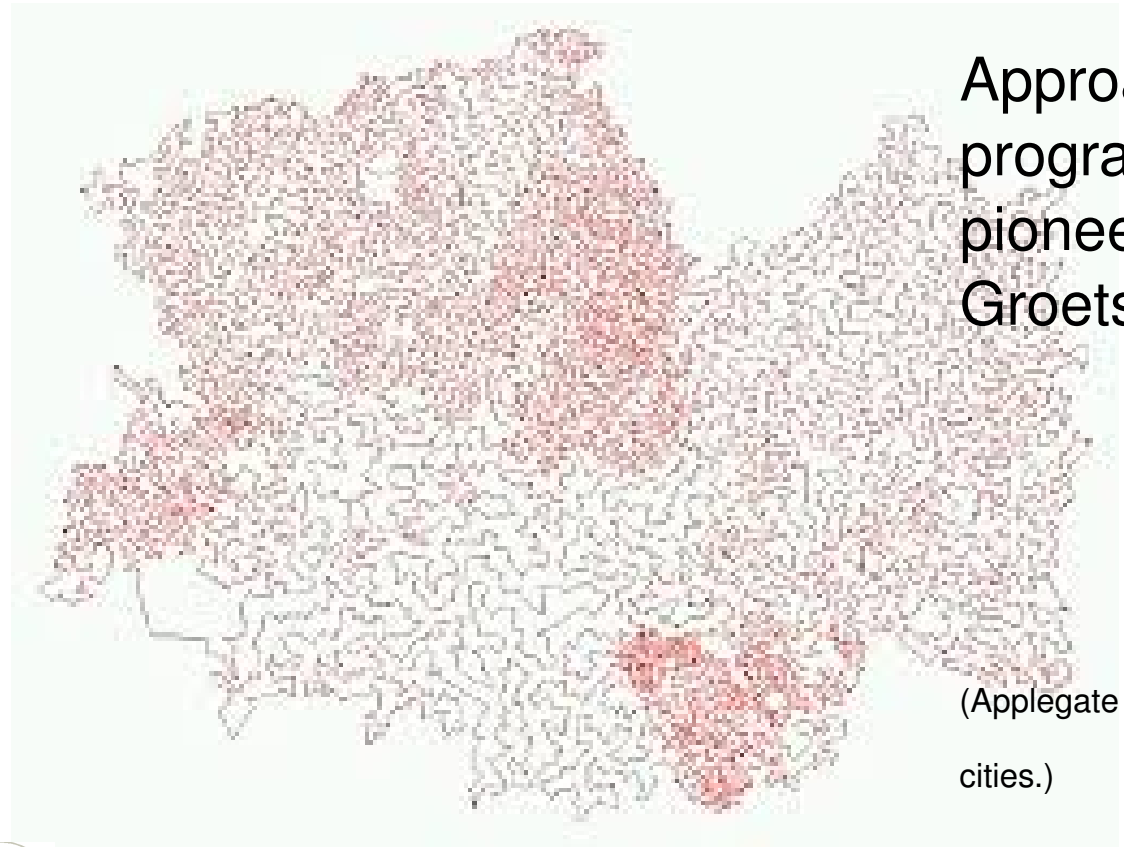


Es gibt viel bessere Approximationsalgorithmen, z.B., Arora und Mitchell.

Jetzt erst recht am Beispiel des Problems des Handlungsreisenden

Gegeben Punkte in der Ebene. Man finde eine kürzeste Rundreise durch alle Punkte.

Exakte Algorithmen für mittelgroße Instanzen.



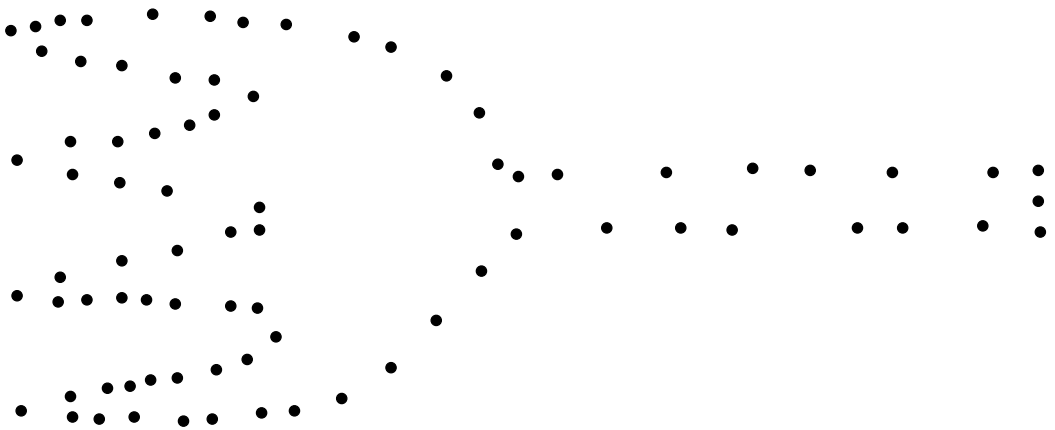
Approach uses linear programming and was pioneered in Berlin (M. Groetschel)

(Applegate et al.: An optimal TSP tour through 85900 cities.)

Jetzt erst recht am Beispiel des Problems des Handlungsreisenden

Gegeben Punkte in der Ebene. Man finde eine kürzeste Rundreise durch alle Punkte.

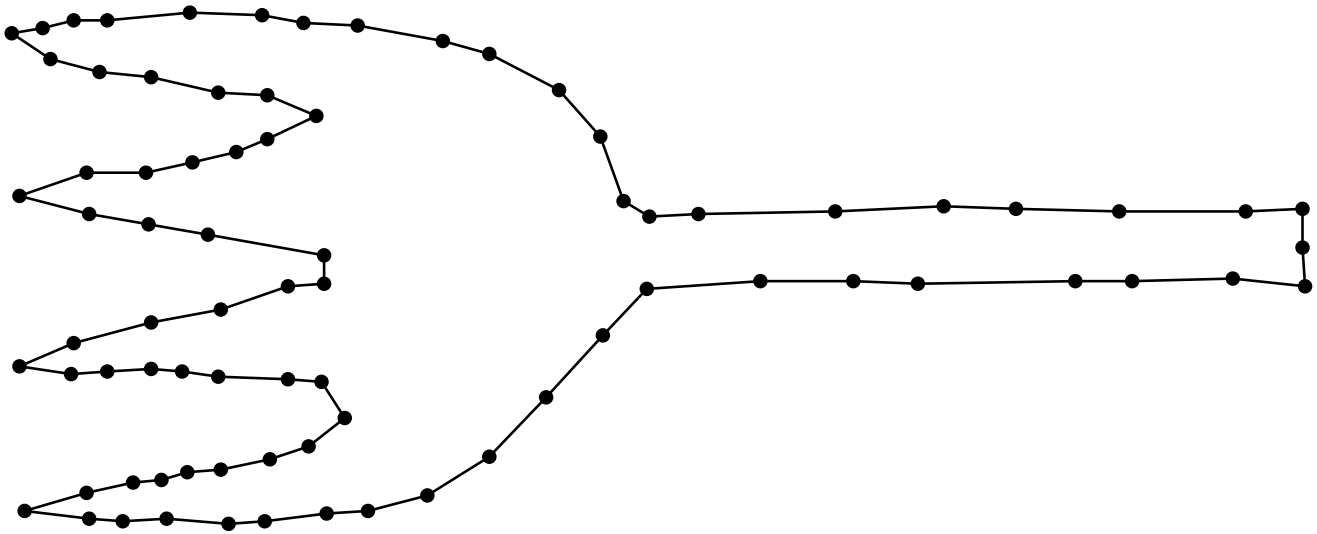
Exakte Algorithmen für Spezialfälle: für Punkte, die von einer gutartigen Kurve kommen, ist TSP in Polynomzeit. Sie sehen nicht nur Punkte, Sie sehen mehr. (Althaus/Mehlhorn, 2002).



Jetzt erst recht am Beispiel des Problems des Handlungsreisenden

Gegeben Punkte in der Ebene. Man finde eine kürzeste Rundreise durch alle Punkte.

Exakte Algorithmen für Spezialfälle: für Punkte, die von einer gutartigen Kurve kommen, ist TSP in Polynomzeit (Althaus/Mehlhorn, 2002).



Zusammenfassung

- P = Menge der Probleme, bei denen eine Lösung in effizient gefunden werden kann.
- NP = Menge der Probleme, bei denen ein Lösungskandidat effizient überprüft werden kann.
- P = NP genau wenn es einen polynomiellen Algorithmus für das Erfüllbarkeitsproblem der Aussagenlogik (SATisfiability Problem) gibt.
- Nicht nur SAT ist NP-vollständig, sondern auch mehrere Tausend anderer Probleme.
- P = NP, eines der großen offenen Probleme der Informatik/Mathematik (Clay Prize)
- Falls P $\neq$ NP, dann ...
- Falls P $=$ NP, dann ...

Informatik für Alle: Ideen und Konzepte der Informatik

Informatik hat die Welt verändert und wird sie weiter verändern.

Wie wir arbeiten, wie wir kommunizieren, wie wir spielen und unsere Freizeit verbringen, wie unsere Wirtschaft funktioniert, was unsere Gesellschaft zusammenhält.

Die Geschwindigkeit der Änderungen wird eher zunehmen.

Jede Bürgerin/jeder Bürger sollte Grundwissen in Informatik haben.

Meine Antwort: Videokurs Ideen und Konzepte der Informatik

Informatik für Alle: Ideen und Konzepte der Informatik

Informatik hat die Welt verändert und wird sie weiter verändern.

Wie wir arbeiten, wie wir kommunizieren, wie wir spielen und unsere Freizeit verbringen, wie unsere Wirtschaft funktioniert, was unsere Gesellschaft zusammenhält.

Die Geschwindigkeit der Änderungen wird eher zunehmen.

Jede Bürgerin/jeder Bürger sollte Grundwissen in Informatik haben.

Meine Antwort: Videokurs Ideen und Konzepte der Informatik

Informatik für Alle: Ideen und Konzepte der Informatik

Informatik hat die Welt verändert und wird sie weiter verändern.

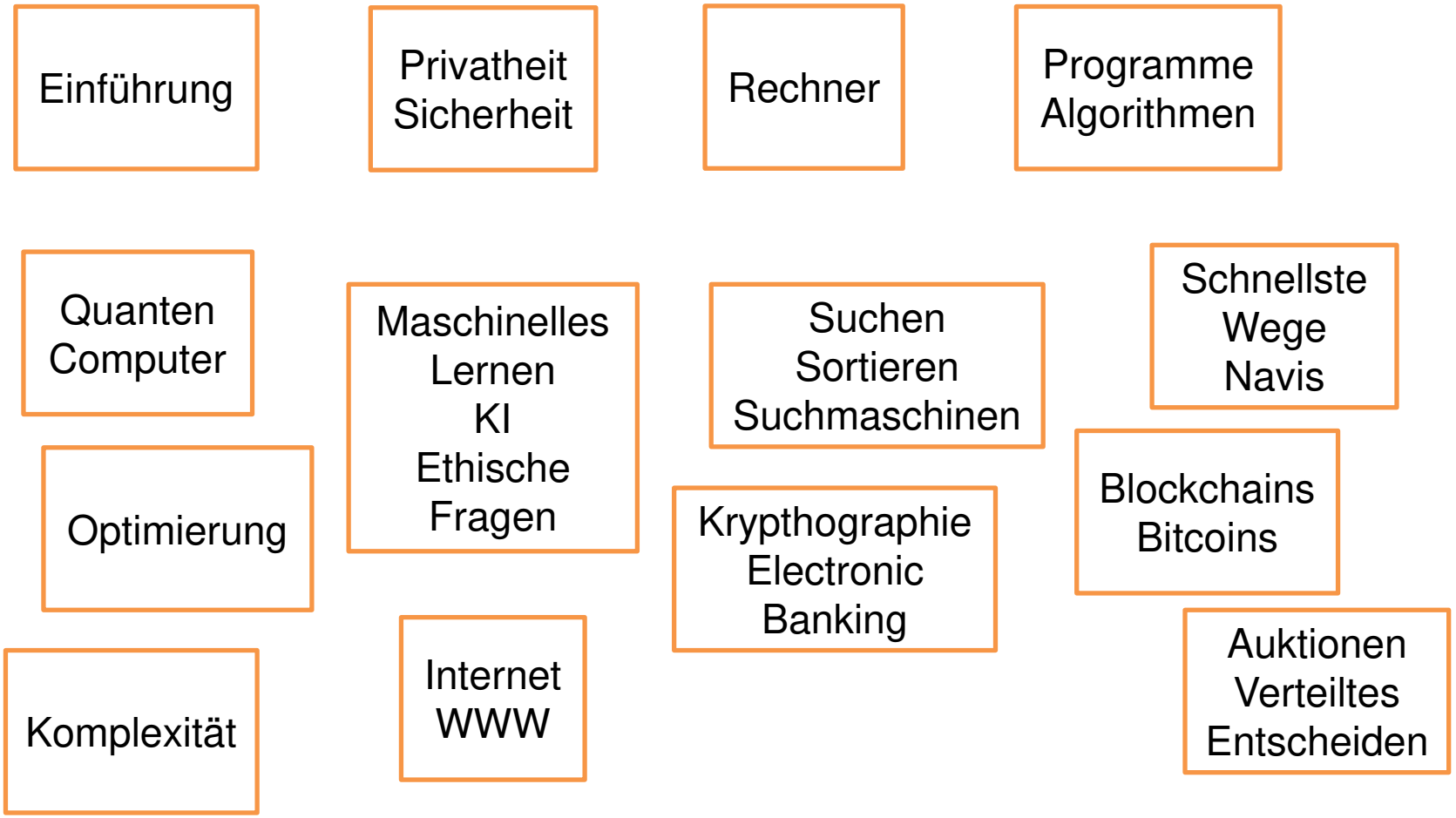
Wie wir arbeiten, wie wir kommunizieren, wie wir spielen und unsere Freizeit verbringen, wie unsere Wirtschaft funktioniert, was unsere Gesellschaft zusammenhält.

Die Geschwindigkeit der Änderungen wird eher zunehmen.

Jede Bürgerin/jeder Bürger sollte Grundwissen in Informatik haben.

Meine Antwort: Videokurs Ideen und Konzepte der Informatik

Themen



Videos sind insgesamt 16 x 90 Minuten.

Umsetzung an der UdS und bei Iversity

- An der UdS
  - Kurs Ideen und Konzepte der Informatik, seit 2016
  - 90 Hörer im WS 21/22
    30% Juristen, 30% Lehrer, 30% MINT-Studenten, ...
- Bei Iversity
  - Iversity = Tochter von Springer Nature, Videokurse
  - insgesamt 1000+ Anmeldungen
  - Folien, Videos, Übungen, Interaktion zwischen Teilnehmern
  - Kurse sind kostenpflichtig, aber

    Mitglieder der UdS und alle heutigen Zuhörer haben freien Zugang über meine Homepage
    `https://people.mpi-inf.mpg.de/~mehlhorn/` oder nach Kurt Mehlhorn googlen.

Freier Zugang

Danke fürs Zuhören