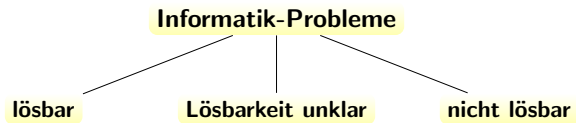


Grenzen algorithmischer Verfahren Theorie und Alltag

PD Dr. Matthias Wendlandt

Institut für Informatik
Justus-Liebig-Universität Gießen

Probleme der Informatik



**Was können wir mit unserem Computer
berechnen?**

Wo liegen die Grenzen?

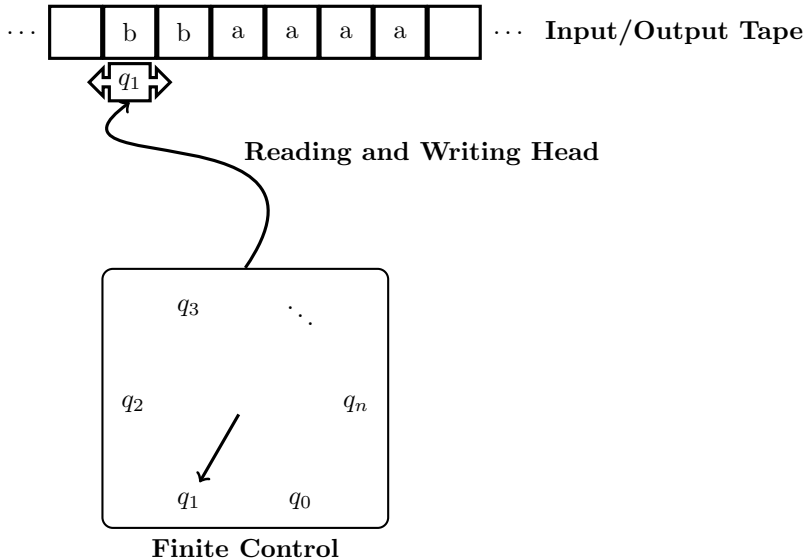
Wie können wir Probleme formal beschreiben?
Was kann ein Mensch berechnen?

Alan Mathison Turing

- ▶ Alan Mathison Turing promovierte 1938 in Princeton und unterrichtete danach am King's College in Cambridge.
- ▶ In seiner berühmtesten Arbeit „On computable numbers, with an application to the Entscheidungsproblem“ stellt er das Konzept der Turing Maschine vor und zeigt auch die Nicht-Berechenbarkeit verschiedener Probleme.



Turingmaschine



Berechenbarkeit

Eine Funktion f ist berechenbar, wenn es eine Turingmaschine gibt, die, auf eine Eingabe x angesetzt, nach endlich vielen Schritten stoppt und das Ergebnis $f(x)$ ausgibt.

- Es gab viele Versuche andere Berechnungsmodelle zu entwickeln (While, Loop, primitiv-rekursiv, μ -rekursiv, ...).

Berechenbarkeit

Eine Funktion f ist berechenbar, wenn es eine Turingmaschine gibt, die, auf eine Eingabe x angesetzt, nach endlich vielen Schritten stoppt und das Ergebnis $f(x)$ ausgibt.

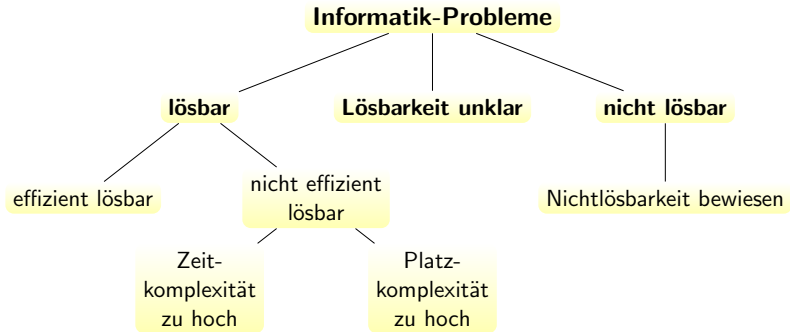
- ▶ Es gab viele Versuche andere Berechnungsmodelle zu entwickeln (While, Loop, primitiv-rekursiv, μ -rekursiv, ...).
- ▶ Es zeigte sich, dass kein Modell „mächtiger“ ist als die Turingmaschine.

Berechenbarkeit

Eine Funktion f ist berechenbar, wenn es eine Turingmaschine gibt, die, auf eine Eingabe x angesetzt, nach endlich vielen Schritten stoppt und das Ergebnis $f(x)$ ausgibt.

- ▶ Es gab viele Versuche andere Berechnungsmodelle zu entwickeln (While, Loop, primitiv-rekursiv, μ -rekursiv, ...).
- ▶ Es zeigte sich, dass kein Modell „mächtiger“ ist als die Turingmaschine.
- ▶ Das Halteproblem ist nicht lösbar.

Probleme der Informatik



Partitionsproblem

Gegeben seien 38 Gegenstände mit den folgenden Massen (in Gramm):

14175, 15055, 16616, 17495, 18072, 19390, 19731, 22161, 23320,
23717, 26343, 28725, 29127, 32257, 40020, 41867, 43155, 46298,
56734, 57176, 58306, 61848, 65825, 66042, 68634, 69189, 72936,
74287, 74537, 81942, 82027, 82623, 82802, 82988, 90467, 97042
97507, 99564

Alle Gegenstände zusammen wiegen 2000000 Gramm. Die Gegenstände sollen auf zwei Container verteilt werden. Jeder Container hat eine Maximallast von 1000000 Gramm. Ist dies möglich?

Partitionsproblem

Gegeben eine Menge von n Zahlen. Gibt es eine Aufteilung dieser Menge in zwei Teilmengen P und Q , so dass die Summe der Zahlen in P gleich der Summe der Zahlen in Q ist?

Partitionsproblem

Gegeben eine Menge von n Zahlen. Gibt es eine Aufteilung dieser Menge in zwei Teilmengen P und Q , so dass die Summe der Zahlen in P gleich der Summe der Zahlen in Q ist?

$$S = \{3, 1, 1, 2, 2, 1\}$$

$$S_1 = \{1, 1, 1, 2\}, S_2 = \{2, 3\}$$

$$1 + 1 + 1 + 1 + 2 = 3 + 2 = 5$$

Partitionsproblem

Gegeben eine Menge von n Zahlen. Gibt es eine Aufteilung dieser Menge in zwei Teilmengen P und Q , so dass die Summe der Zahlen in P gleich der Summe der Zahlen in Q ist?

$$S = \{3, 1, 1, 2, 2, 1\}$$

$$S_1 = \{1, 1, 1, 2\}, S_2 = \{2, 3\}$$

$$1 + 1 + 1 + 1 + 2 = 3 + 2 = 5$$

- Das Problem kann mit den bisher bekannten Techniken für große Probleminstanzen algorithmisch in vernünftiger Zeit nicht gelöst werden.

Komplexitätstheorie

- ▶ 1962 legten **Juris Hartmanis und Richard Stearns** den Grundstein für das Forschungsgebiet **Komplexitätstheorie**.

Komplexitätstheorie

- ▶ 1962 legten **Juris Hartmanis und Richard Stearns** den Grundstein für das Forschungsgebiet **Komplexitätstheorie**.
- ▶ In ihrem Papier „**On the computational complexity of algorithms**“ klassifizierten sie Probleme anhand ihres **Zeit- und Platzbedarfs**. Hierfür erhielten sie 1993 den **Turing Award**.

Komplexitätstheorie

- ▶ 1962 legten **Juris Hartmanis und Richard Stearns** den Grundstein für das Forschungsgebiet **Komplexitätstheorie**.
- ▶ In ihrem Papier „**On the computational complexity of algorithms**“ klassifizierten sie Probleme anhand ihres **Zeit- und Platzbedarfs**. Hierfür erhielten sie 1993 den **Turing Award**.
- ▶ Alan Cobham lieferte den entscheidenden Beitrag zur Definition des Begriffs „**effizient lösbare Probleme**“ in seinem Papier „**The intrinsic computational difficulty of functions**“ über die **Definition der Komplexitätsklasse P**.

Komplexität – Was ist das?

Wie komplex ist ein Problem?

- ▶ So komplex wie der bestmögliche, das Problem lösende Algorithmus.
- ▶ Als Maßstab wird das Worst-Case Szenario verwendet.

Komplexität – Was ist das?

Wie komplex ist ein Problem?

- ▶ So komplex wie der **bestmögliche, das Problem lösende Algorithmus**.
- ▶ Als Maßstab wird das **Worst-Case Szenario** verwendet.

Wie wird Komplexität gemessen?

- ▶ **Zeit**: Zählen möglichst **elementarer Operationen**.
- ▶ **Platz**: Zählen der **benötigten Speicherplätze**.

Wie wird Komplexität gemessen?

```
void selectionSort(int arr[], int n){
    int i, j, t, min;
    for (i = 0; i < n-1; i++){
        min = i;
        for (j = i+1; j < n; j++){
            if (arr[j] < arr[min]){
                min = j;
            }
        }
        t=arr[i];
        arr[i]=arr[min];
        arr[min]=t;
    }
    cout >> "Array sortiert";
}
```

Wie wird Komplexität gemessen?

Abstraktionen

- ▶ Nur das **asymptotische Verhalten auf großen Problemen** ist interessant, da genaue Betrachtungen nicht möglich sind.
- ▶ **Additive Konstanten** werden nicht berücksichtigt.
- ▶ **Konstante Faktoren** werden nicht berücksichtigt.

Die Komplexitätsklasse P

P ist die Klasse derjenigen Probleme, für die es einen deterministischen Polynomialzeitalgorithmus gibt.

Die Komplexitätsklasse NP

NP ist die Klasse derjenigen Probleme, für die es einen nichtdeterministischen Polynomialzeitalgorithmus gibt.

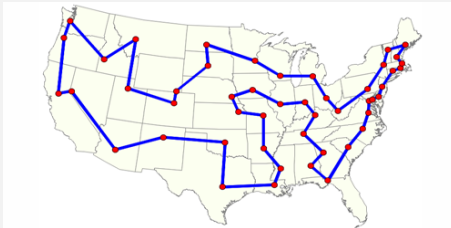
Travelling Salesman Problem

Ein Handlungsreisender soll n Städte nacheinander genau einmal besuchen und wieder zu seinem Ausgangspunkt zurückkehren. Gibt es einen Rundweg, der höchstens die Länge l_{max} hat?

Das Travelling Salesman Problem ist NP-vollständig.

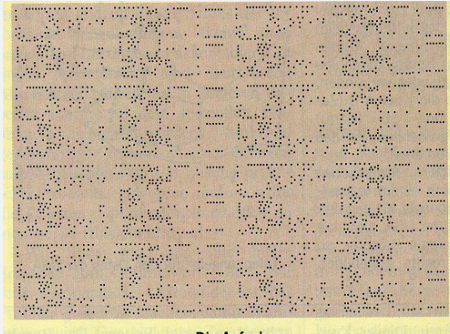
Beispiele Travelling Salesman Problem

Die bekannteste Anwendung ist die des Handlungsreisenden.



Beispiele Travelling Salesman Problem

Das Problem findet sich auch in der Industrie, beispielsweise beim Bohren von Leiterplatten, wieder.



$P \neq NP?$

Stephen A. Cook

- ▶ Stephen Arthur Cook (*14. Dezember 1939 in Buffalo, New York) ist Professor der Informatik an der University of Toronto in Kanada.
- ▶ Cook wurde in der theoretischen Informatik berühmt durch den Satz von Cook: „SAT ist NP-vollständig“ .
- ▶ 1982 bekam er für diese Entdeckung den Turing Award.
- ▶ Cooks erster Doktorand war Walter Savitch.



3-Sat Problem

„Wenn es nicht regnet und die Sonne scheint, dann ist schönes Wetter. “

3-Sat Problem

„Wenn es nicht regnet und die Sonne scheint, dann ist schönes Wetter. “

„Wenn das Einkommen > 4000 und keine Schulden, dann kreditwürdig.“

3-Sat Problem

„Wenn es nicht regnet und die Sonne scheint, dann ist schönes Wetter. “

„Wenn das Einkommen > 4000 und keine Schulden, dann kreditwürdig.“

„Wenn die Temperatur niedrig, dann Öffnung des Heizungsventils groß.“

3-Sat Problem

„Wenn es nicht regnet und die Sonne scheint, dann ist schönes Wetter. “

„Wenn das Einkommen > 4000 und keine Schulden, dann kreditwürdig.“

„Wenn die Temperatur niedrig, dann Öffnung des Heizungsventils groß.“

Die Bedeutung einer solchen Formel ist,

Wenn **A** wahr (bewiesen) ist,
dann schließe, dass auch **B** wahr ist.

3-Sat Problem

„Wenn es nicht regnet und die Sonne scheint, dann ist schönes Wetter. “

„Wenn das Einkommen > 4000 und keine Schulden, dann kreditwürdig.“

„Wenn die Temperatur niedrig, dann Öffnung des Heizungsventils groß.“

Die Bedeutung einer solchen Formel ist,

Wenn **A** wahr (bewiesen) ist,
dann schließe, dass auch **B** wahr ist.

$$\overline{A} \wedge B$$

3-SAT Problem

Gegeben eine in **konjunktiver Normalform** vorliegende **ausagenlogische Formel** F , die **höchstens drei Literale** enthält. Ist F erfüllbar?

3-SAT Problem

Gegeben eine in **konjunktiver Normalform** vorliegende **ausagenlogische Formel** F , die **höchstens drei Literale** enthält. Ist F erfüllbar?

$$F = (x_1 \vee \overline{x_2} \vee \overline{x_3}) \wedge (x_2 \vee x_3 \vee \overline{x_4}) \wedge (\overline{x_1} \vee x_3)$$

3-SAT Problem

Gegeben eine in **konjunktiver Normalform** vorliegende **aus-sagenlogische Formel** F , die **höchstens drei Literale** enthält. Ist F erfüllbar?

$$F = (x_1 \vee \overline{x_2} \vee \overline{x_3}) \wedge (x_2 \vee x_3 \vee \overline{x_4}) \wedge (\overline{x_1} \vee x_3)$$

$$x_1 = 1, x_2 = 1, x_3 = 1, x_4 = 1$$

3-SAT Problem

Gegeben eine in **konjunktiver Normalform** vorliegende **ausagenlogische Formel** F , die **höchstens drei Literale** enthält. Ist F erfüllbar?

$$F = (x_1 \vee \overline{x_2} \vee \overline{x_3}) \wedge (x_2 \vee x_3 \vee \overline{x_4}) \wedge (\overline{x_1} \vee x_3)$$

$$x_1 = 1, x_2 = 1, x_3 = 1, x_4 = 1$$

$$F = (1 \vee 0 \vee 0) \wedge (1 \vee 1 \vee 0) \wedge (0 \vee 1)$$

3-SAT Problem

Gegeben eine in **konjunktiver Normalform** vorliegende **aus-sagenlogische Formel** F , die **höchstens drei Literale** enthält. Ist F erfüllbar?

$$F = (x_1 \vee \overline{x_2} \vee \overline{x_3}) \wedge (x_2 \vee x_3 \vee \overline{x_4}) \wedge (\overline{x_1} \vee x_3)$$

$$x_1 = 1, x_2 = 1, x_3 = 1, x_4 = 1$$

$$F = (1 \vee 0 \vee 0) \wedge (1 \vee 1 \vee 0) \wedge (0 \vee 1)$$

- Das Problem ist NP-vollständig.

P-NP – Geschichte

- ▶ 1971 zeigte er in dem Paper
„The Complexity of Theorem Proving Procedures“
die Reduktion einer nichtdeterministischen Turingmaschine,
die in polynomieller Zeit arbeitet, auf das SAT Problem.

P-NP – Geschichte

- ▶ 1971 zeigte er in dem Paper

„The Complexity of Theorem Proving Procedures“

die Reduktion einer nichtdeterministischen Turingmaschine, die in polynomieller Zeit arbeitet, auf das SAT Problem.

- ▶ Das Kernstück des Beweises ist der Nachweis der NP-Härte durch die Konstruktion einer logischen Formel für eine gegebene polynomiell beschränkte Turingmaschine.
- ▶ Variablen: $\text{zust}_{t,z}$, $\text{pos}_{t,i}$, $\text{band}_{t,i,a}$
- ▶ $F = R \wedge A \wedge \ddot{U}_1 \wedge \ddot{U}_2 \wedge E$

P-NP – Geschichte

- ▶ 1971 zeigte er in dem Paper

„The Complexity of Theorem Proving Procedures“

die Reduktion einer nichtdeterministischen Turingmaschine, die in polynomieller Zeit arbeitet, auf das SAT Problem.

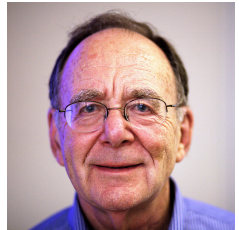
- ▶ Das Kernstück des Beweises ist der Nachweis der NP-Härte durch die Konstruktion einer logischen Formel für eine gegebene polynomiell beschränkte Turingmaschine.
 - ▶ Variablen: $\text{zust}_{t,z}$, $\text{pos}_{t,i}$, $\text{band}_{t,i,a}$
 - ▶ $F = R \wedge A \wedge \ddot{U}_1 \wedge \ddot{U}_2 \wedge E$
- ▶ 1973 entwickelte Leonid Levin unabhängig eine Theorie der NP-Vollständigkeit, die aber lange unberücksichtigt blieb.

P-NP – Geschichte

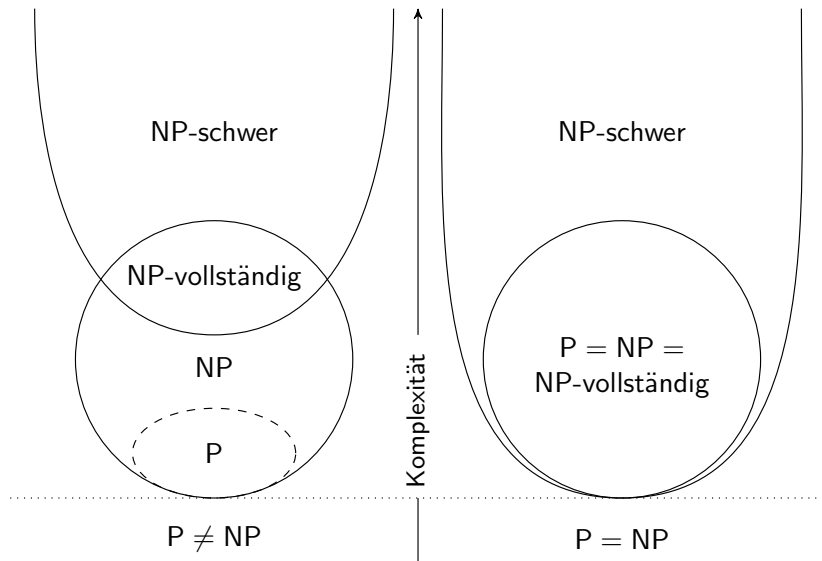
- ▶ Im Jahr darauf wurde diese Arbeit von Richard M. Karp erweitert. Er wies 21 NP-vollständige Probleme aus verschiedenen Bereichen der Mathematik und Informatik nach.

Richard M. Karp

- ▶ Richard Manning Karp (*3. Januar 1935 in Boston) ist ein amerikanischer Informatiker.
- ▶ 1968 wurde er Professor für Informatik, Mathematik und Operations Research an der [University of California](#), Berkeley.
- ▶ Er erhielt 1985 den [Turing Award](#).



NP-Vollständigkeit



Wieso ist die Thematik so spannend?

- ▶ Es beschreibt die momentane Grenze von dem, was Computer können und was nicht.
- ▶ $P=NP$ hätte unvorhersehbare Auswirkungen.
- ▶ $P \neq NP$? ist ein Millennium-Problem und beschäftigt die Menschen schon seit Jahrhunderten.

Wieso ist die Thematik so spannend?

- ▶ Es beschreibt die momentane Grenze von dem, was Computer können und was nicht.
- ▶ $P=NP$ hätte unvorhersehbare Auswirkungen.
- ▶ $P \neq NP$? ist ein Millennium-Problem und beschäftigt die Menschen schon seit Jahrhunderten.
 - ▶ Als Millennium-Probleme bezeichnet man die im Jahr 2000 vom Clay Mathematics Institute (CMI) in Cambridge (Massachusetts) in einer Liste aufgezählten ungelösten Probleme der Mathematik.
 - ▶ Das Institut hat für die Lösung eines der sieben Probleme ein Preisgeld von jeweils einer Million US-Dollar ausgelobt.

Probleme in P

- ▶ Sortieren
- ▶ Kürzeste Wege
- ▶ Minimaler Spannbaum
- ▶ Graphzusammenhang
- ▶ Maximaler Fluss
- ▶ Maximum Matching
- ▶ Lineare Programmierung
- ▶ Größter Gemeinsamer Teiler
- ▶ Primzahltest

Übersicht Komplexitätstheorie - NP

- ▶ 3-Sat Problem
- ▶ Traveling Salesman Problem
- ▶ Partitionsproblem
- ▶ Rucksackproblem
- ▶ Cliquenproblem
- ▶ Hamiltonkreis Problem
- ▶ Graphfärbungsproblem
- ▶ ...

Rucksack Problem

Gegeben k natürliche Zahlen $S = \{a_1, a_2, \dots, a_k\}$ und eine natürliche Zahl b . Gibt es eine Teilmenge $R \subseteq S$, so dass die Summe aller Zahlen in R b ergibt?

Rucksack Problem

Gegeben k natürliche Zahlen $S = \{a_1, a_2, \dots, a_k\}$ und eine natürliche Zahl b . Gibt es eine Teilmenge $R \subseteq S$, so dass die Summe aller Zahlen in R b ergibt?

$$S = \{3, 6, 1, 7, 3, 8\}, b = 19$$

Rucksack Problem

Gegeben k natürliche Zahlen $S = \{a_1, a_2, \dots, a_k\}$ und eine natürliche Zahl b . Gibt es eine Teilmenge $R \subseteq S$, so dass die Summe aller Zahlen in R b ergibt?

$$S = \{3, 6, 1, 7, 3, 8\}, b = 19$$

$$1 + 7 + 3 + 8$$

Rucksack Problem

Gegeben k natürliche Zahlen $S = \{a_1, a_2, \dots, a_k\}$ und eine natürliche Zahl b . Gibt es eine Teilmenge $R \subseteq S$, so dass die Summe aller Zahlen in R b ergibt?

$$S = \{3, 6, 1, 7, 3, 8\}, b = 19$$

$$1 + 7 + 3 + 8$$

- Das Problem ist NP-vollständig.

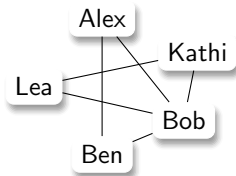
Rucksack Problem

- ▶ Ein Dieb bricht in ein Haus ein. Er hat einen Rucksack dabei. Wie kann er den maximalen Gewinn erzielen?
- ▶ Die Fragestellung findet Anwendung in der Logistik. Gegeben ein Frachtflugzeug mit einer Maximallast. Bei einer Beladung soll die maximale Last aufgenommen werden.

Cliquen Problem

Gegeben ein ungerichteter Graph $G = (V, E)$ und eine natürliche Zahl k . Besitzt G eine Clique der Größe k ?

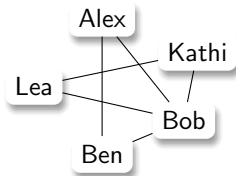
- Eine Clique ist ein Teilgraph, in dem jedes Knotenpaar mit einer Kante verbunden sind.



Cliquen Problem

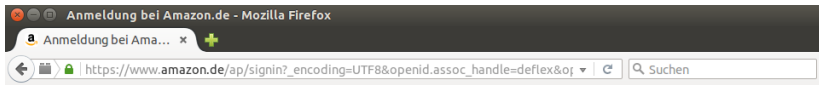
Gegeben ein ungerichteter Graph $G = (V, E)$ und eine natürliche Zahl k . Besitzt G eine Clique der Größe k ?

- Eine Clique ist ein Teilgraph, in dem jedes Knotenpaar mit einer Kante verbunden sind.



- Gibt es eine Clique mit der Größe 4?
- Wie groß ist die größte Clique in Facebook?
- Das Problem ist NP-vollständig.

HTTPS – Public Key



[Mein Konto](#) | [Hilfe](#)

Anmelden

Wie lautet Ihre E-Mail-Adresse oder Mobiltelefonnummer?

E-Mail oder Mobiltelefonnummer:

Haben Sie ein Passwort für Amazon.de?

☐ **Nein, ich bin ein neuer Kunde.**

☒ **Ja, ich habe ein Passwort:**

[Haben Sie Ihr Passwort vergessen?](#)

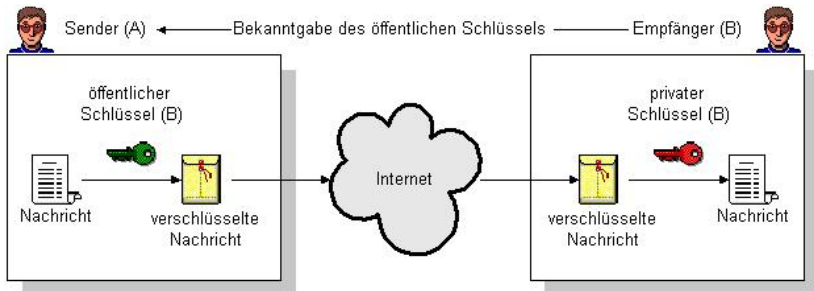
Weiter (über den Sicherheitsserver) 

Mit Ihrer Anmeldung erklären Sie sich mit [Unseren AGB](#), unserer [Datenschutzerklärung](#) sowie den [Bestimmungen zu Cookies & Internet-Werbung](#) einverstanden.

Hilfe zur Anmeldung

Passwort vergessen? [Passwort-Hilfe anfordern](#).

Public Key Verschlüsselung – RSA Algorithmus



Kryptographie – P-NP Problem

- Wähle zufällig zwei Primzahlen $p \neq q$.
 $p = 11, q = 13$

Kryptographie – P-NP Problem

- ▶ Wähle zufällig zwei Primzahlen $p \neq q$.
 $p = 11, q = 13$
- ▶ Berechne den RSA-Modul $N = p \cdot q$.
 $11 \cdot 13 = 143$

Kryptographie – P-NP Problem

- ▶ Wähle zufällig zwei Primzahlen $p \neq q$.
 $p = 11, q = 13$
- ▶ Berechne den RSA-Modul $N = p \cdot q$.
 $11 \cdot 13 = 143$
- ▶ Berechne die Eulersche ϕ -Funktion von N
 $\phi(N) = (p - 1) \cdot (q - 1)$.
 $\phi(143) = (11 - 1) \cdot (13 - 1) = 120$

Kryptographie – P-NP Problem

- ▶ Wähle zufällig zwei Primzahlen $p \neq q$.
 $p = 11, q = 13$
- ▶ Berechne den RSA-Modul $N = p \cdot q$.
 $11 \cdot 13 = 143$
- ▶ Berechne die Eulersche ϕ -Funktion von N
 $\phi(N) = (p - 1) \cdot (q - 1)$.
 $\phi(143) = (11 - 1) \cdot (13 - 1) = 120$
- ▶ Wähle eine zu $\phi(N)$ teilerfremde Zahl e .
 $e = 23$

Kryptographie – P-NP Problem

- ▶ Wähle zufällig zwei Primzahlen $p \neq q$.
 $p = 11, q = 13$
- ▶ Berechne den RSA-Modul $N = p \cdot q$.
 $11 \cdot 13 = 143$
- ▶ Berechne die Eulersche ϕ -Funktion von N
 $\phi(N) = (p - 1) \cdot (q - 1)$.
 $\phi(143) = (11 - 1) \cdot (13 - 1) = 120$
- ▶ Wähle eine zu $\phi(N)$ teilerfremde Zahl e .
 $e = 23$
- ▶ Berechne die Entschlüsselungskomponente d mit
 $e \cdot d \equiv 1 \pmod{\phi(N)}$.
 $23 \cdot 47 \equiv 1 \pmod{120}$

Kryptographie – P-NP Problem

- ▶ Wähle zufällig zwei Primzahlen $p \neq q$.
 $p = 11, q = 13$
- ▶ Berechne den RSA-Modul $N = p \cdot q$.
 $11 \cdot 13 = 143$
- ▶ Berechne die Eulersche ϕ -Funktion von N
 $\phi(N) = (p - 1) \cdot (q - 1)$.
 $\phi(143) = (11 - 1) \cdot (13 - 1) = 120$
- ▶ Wähle eine zu $\phi(N)$ teilerfremde Zahl e .
 $e = 23$
- ▶ Berechne die Entschlüsselungskomponente d mit
 $e \cdot d \equiv 1 \pmod{\phi(N)}$.
 $23 \cdot 47 \equiv 1 \pmod{120}$

Verschlüsseln der Zahl 7: $2 \equiv 7^{23} \pmod{143}$

Entschlüsseln der Zahl 2: $7 \equiv 2^{47} \pmod{143}$

Kryptographie – P-NP Problem

- ▶ Wähle zufällig zwei Primzahlen $p \neq q$.
 $p = 11, q = 13$
- ▶ Berechne den RSA-Modul $N = p \cdot q$.
 $11 \cdot 13 = 143$
- ▶ Berechne die Eulersche ϕ -Funktion von N
 $\phi(N) = (p - 1) \cdot (q - 1)$.
 $\phi(143) = (11 - 1) \cdot (13 - 1) = 120$
- ▶ Wähle eine zu $\phi(N)$ teilerfremde Zahl e .
 $e = 23$
- ▶ Berechne die Entschlüsselungskomponente d mit
 $e \cdot d \equiv 1 \pmod{\phi(N)}$.
 $23 \cdot 47 \equiv 1 \pmod{120}$
- ▶ Das Schwierige beim Entschlüsseln ohne Geheimschlüssel ist das Faktorisieren der Zahl N .
- ▶ Kann ein Algorithmus nichtdeterministisch arbeiten, dann rät er die Zahlen p und q .

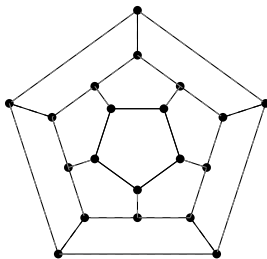
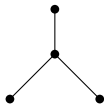
Ist nicht vielleicht doch $P=NP$?

Ein freundschaftlicher Umtrunk – Hamiltonkreis

Hamiltonkreis

Theorem

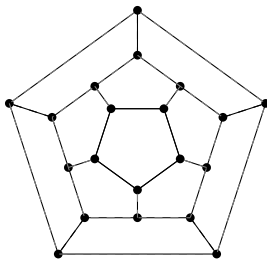
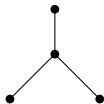
Ein **Hamiltonpfad** besucht alle Knoten eines Graphen genau einmal.
Ein **Hamiltonkreis** kehrt außerdem zum Anfangsknoten zurück.



Hamiltonkreis

Theorem

Ein **Hamiltonpfad** besucht alle Knoten eines Graphen genau einmal.
Ein **Hamiltonkreis** kehrt außerdem zum Anfangsknoten zurück.

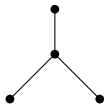


Dieser Graph besitzt
keinen Hamilton-Kreis.

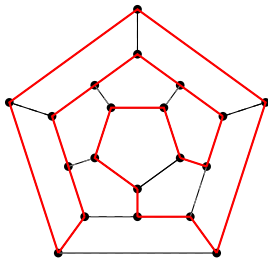
Hamiltonkreis

Theorem

Ein **Hamiltonpfad** besucht alle Knoten eines Graphen genau einmal.
Ein **Hamiltonkreis** kehrt außerdem zum Anfangsknoten zurück.



Dieser Graph besitzt
keinen **Hamilton-Kreis**.

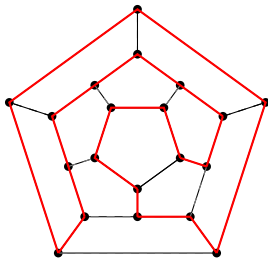
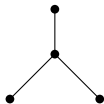


Dieser Graph besitzt
einen **Hamilton-Kreis**.

Hamiltonkreis

Theorem

Ein **Hamiltonpfad** besucht alle Knoten eines Graphen genau einmal.
Ein **Hamiltonkreis** kehrt außerdem zum Anfangsknoten zurück.



Dieser Graph besitzt **keinen Hamilton-Kreis**. Dieser Graph besitzt **einen Hamilton-Kreis**.

► Dieses Problem ist **NP-vollständig**.

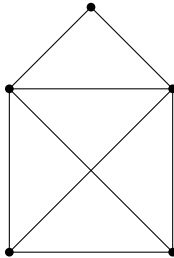
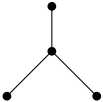
Eulerkreis

- ▶ Eine kleine Veränderung ergibt den Eulerkreis.
- ▶ Kann eine Figur in einem Zug nachgemalt werden?
- ▶ Gegeben eine Inselkette und Verbindungen zwischen diesen Inseln. Kann eine Frachtschiff jede einzelne Verbindung nacheinander abfahren, ohne eine Verbindung doppelt zu verwenden?

Eulerkreis

Theorem

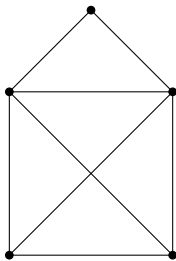
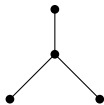
Ein **Eulerpfad** durchläuft alle Kanten eines Graphen genau einmal, ein **Eulerkreis** kehrt außerdem zum Anfangsknoten zurück.



Eulerkreis

Theorem

Ein **Eulerpfad** durchläuft alle Kanten eines Graphen genau einmal, ein **Eulerkreis** kehrt außerdem zum Anfangsknoten zurück.



- Dieses **Problem** kann durch **Testen** der Eingangsgrade in $O(n^2)$ Zeit gelöst werden.

Sat Problem

3-Sat Problem

Gegeben eine in konjunktiver Normalform vorliegende aussagenlogische Formel F , die höchstens drei Literale enthält. Ist F erfüllbar?

- Dieses Problem ist NP-vollständig.

Sat Problem

3-Sat Problem

Gegeben eine in konjunktiver Normalform vorliegende aussagenlogische Formel F , die höchstens **drei** Literale enthält. Ist F erfüllbar?

- Dieses Problem ist **NP-vollständig**.

2-Sat Problem

Gegeben eine in konjunktiver Normalform vorliegende aussagenlogische Formel F , die höchstens **zwei** Literale enthält. Ist F erfüllbar?

- Dieses Problem ist in **P**.

Das zweite LBA Problem

- ▶ Das Problem $\text{NSPACE} \stackrel{?}{=} \text{co-NSPACE}$ wurde 1964 von Kuroda formuliert.
- ▶ Lange Zeit war man sich sicher, dass $\text{NSPACE} \neq \text{co-NSPACE}$.
- ▶ 1987 haben Immerman und Szelepcsényi unabhängig voneinander bewiesen, dass $\text{NSPACE} = \text{co-NSPACE}$.

Primzahlen

- ▶ Lange Zeit glaubte man, dass das Testen einer Primzahl nicht in P läge und man damit $P \neq NP$ zeigen kann.
- ▶ 2002 wurde jedoch von Agrawal, Kayal und Saxena ein Algorithmus in $O(n^6)$ entwickelt, der für eine Zahl testet, ob sie eine Primzahl ist.

Nicht immer ist der optimale Weg der sinnvollste

Traveling Salesman Problem – Nearest-Insertion-Heuristik

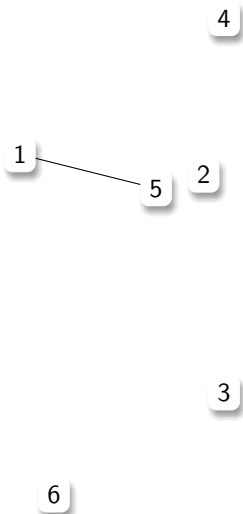
- ▶ Das Travelling Salesman Problem ist NP-vollständig.
- ▶ Bei Vorliegen der Dreiecksungleichung gibt es Algorithmen, die in $O(n^2)$ eine maximal 1,5 fach schlechtere Lösung berechnen.

Approximationsverfahren – TSP

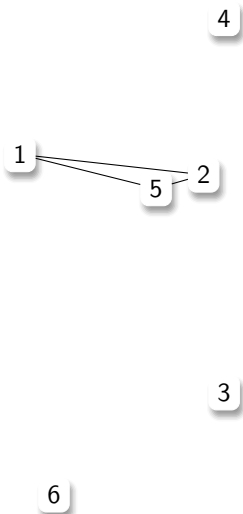
Bei Vorliegen der **Dreiecksungleichung** gibt es Algorithmen, die wesentlich schneller einen Weg berechnen, der jedoch nicht notwendigerweise optimal ist. Ein Beispiel ist die Nearest-Insertion Heuristik:

- ▶ **Wähle** einen **Knoten** mit der **geringsten Entfernung** zu einem **Knoten** der schon **konstruierten Teilroute**.
- ▶ **Baue** diesen **Knoten** in die **vorhandene Teilroute** ein, so dass die **geringste Verlängerung der bisherigen Teilroute** entsteht.
- ▶ Bei Vorliegen der **Dreiecksungleichung** kann die Länge der gefundenen Rundreise aber nicht schlechter als das **Doppelte der Länge** einer optimalen Rundreise sein. Die Laufzeit liegt in $O(n^2)$.

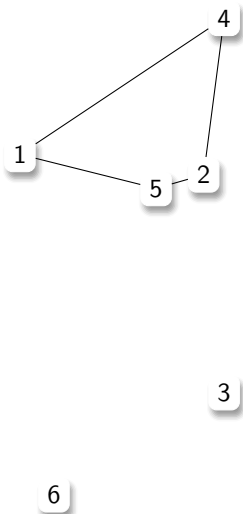
Traveling Salesman Problem – Nearest-Insertion Heuristik



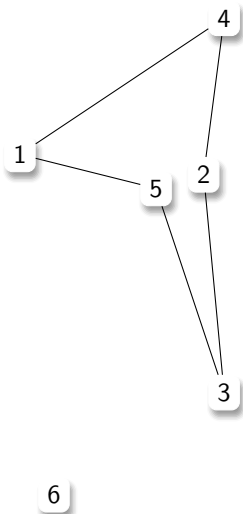
Traveling Salesman Problem – Nearest-Insertion Heuristik



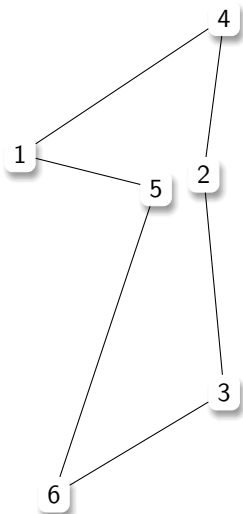
Traveling Salesman Problem – Nearest-Insertion Heuristik



Traveling Salesman Problem – Nearest-Insertion Heuristik



Traveling Salesman Problem – Nearest-Insertion Heuristik



Bin Packing Problem

Gegeben seien n Objekte der Größen s_1, \dots, s_n mit $0 < s_i \leq 1$, für $1 \leq i \leq n$. Gesucht ist die kleinstmögliche Anzahl von Kisten (Bins) der Größe 1, mit der alle Objekte verpackt werden können.

Dieses Problem ist NP-vollständig

Bin Packing Problem

Der folgende Approximationsalgorithmus **First Fit Decreasing** löst das Bin Packing Problem schneller, wenn auch nicht notwendigerweise optimal:

- ▶ **Sortiere die Objekte** zunächst nach abnehmender Größe.
- ▶ **Verpacke die Objekte** in **absteigender Reihenfolge**. Wähle dabei immer den **ersten Behälter**, in dem noch **genug Platz** ist.
- ▶ Sind alle **Behälter voll**, füge einen **neuen hinzu**.

Der obige Approximationsalgorithmus *First Fit Decreasing* löst das Problem asymptotisch mit **Gütegarantie** $\frac{11}{9}$. Die **Laufzeit** ist $O(n \cdot \log n)$.

Danke für die Aufmerksamkeit!