Graphen und diskrete Optimierung

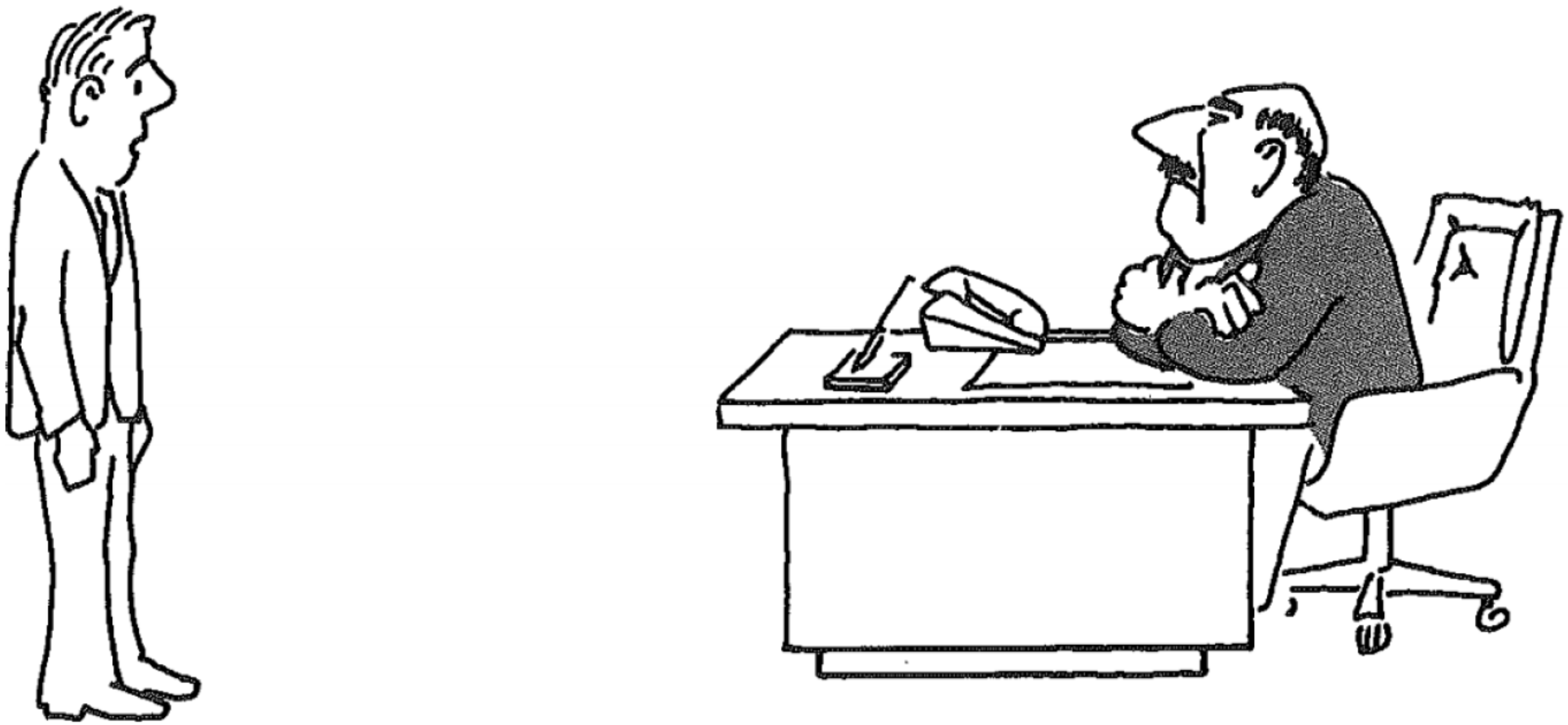
im Bachelorstudiengang 'Informatik und Nachhaltigkeit'

Komplexität: P, NP, NP-vollständig und NP-schwer

Marie Schmidt

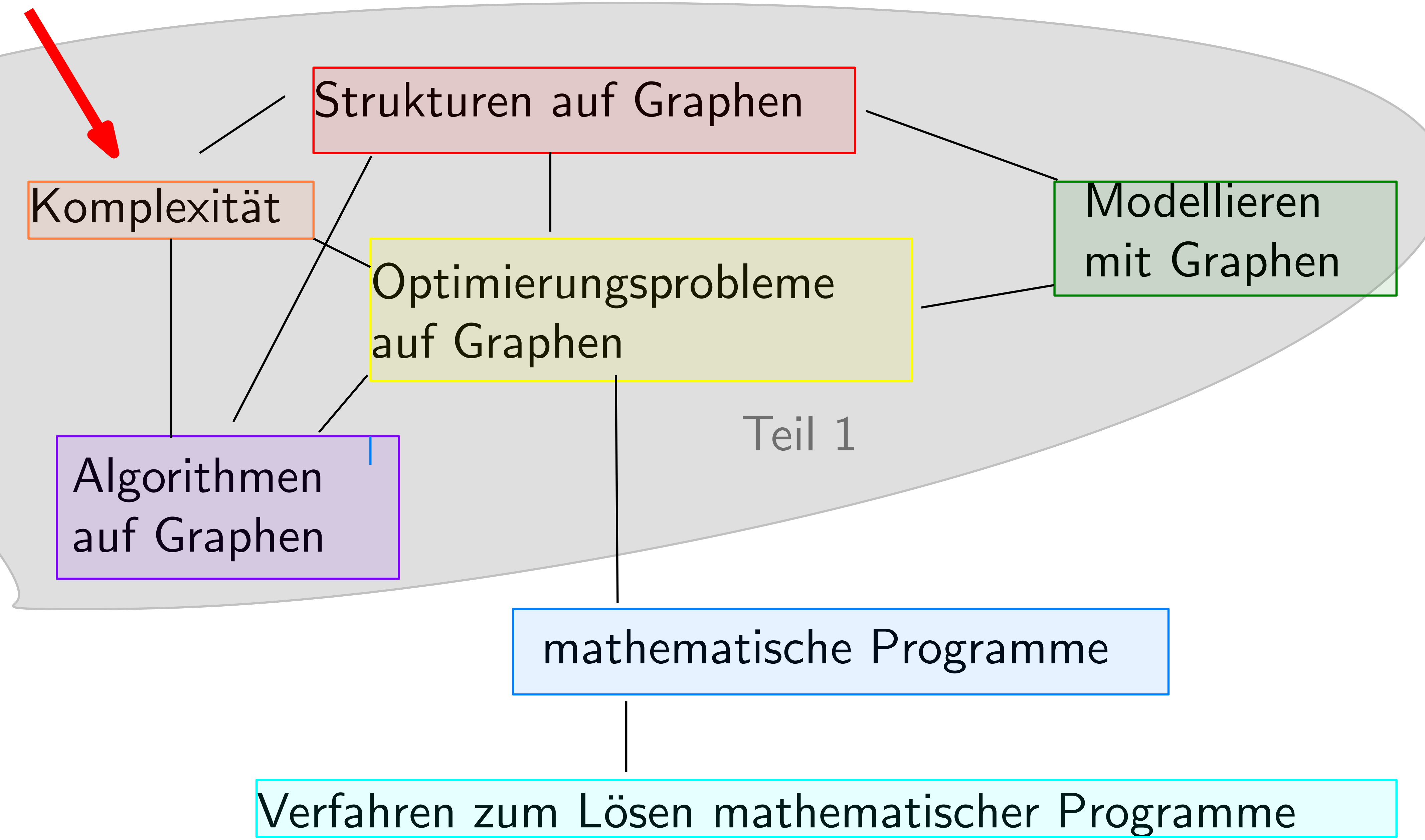
24.05.2023

Unangenehme Situation. . .



“I can’t find an efficient algorithm, I guess I’m just too dumb.”

Worum soll er hier gehen?



Vorlesungsübersicht

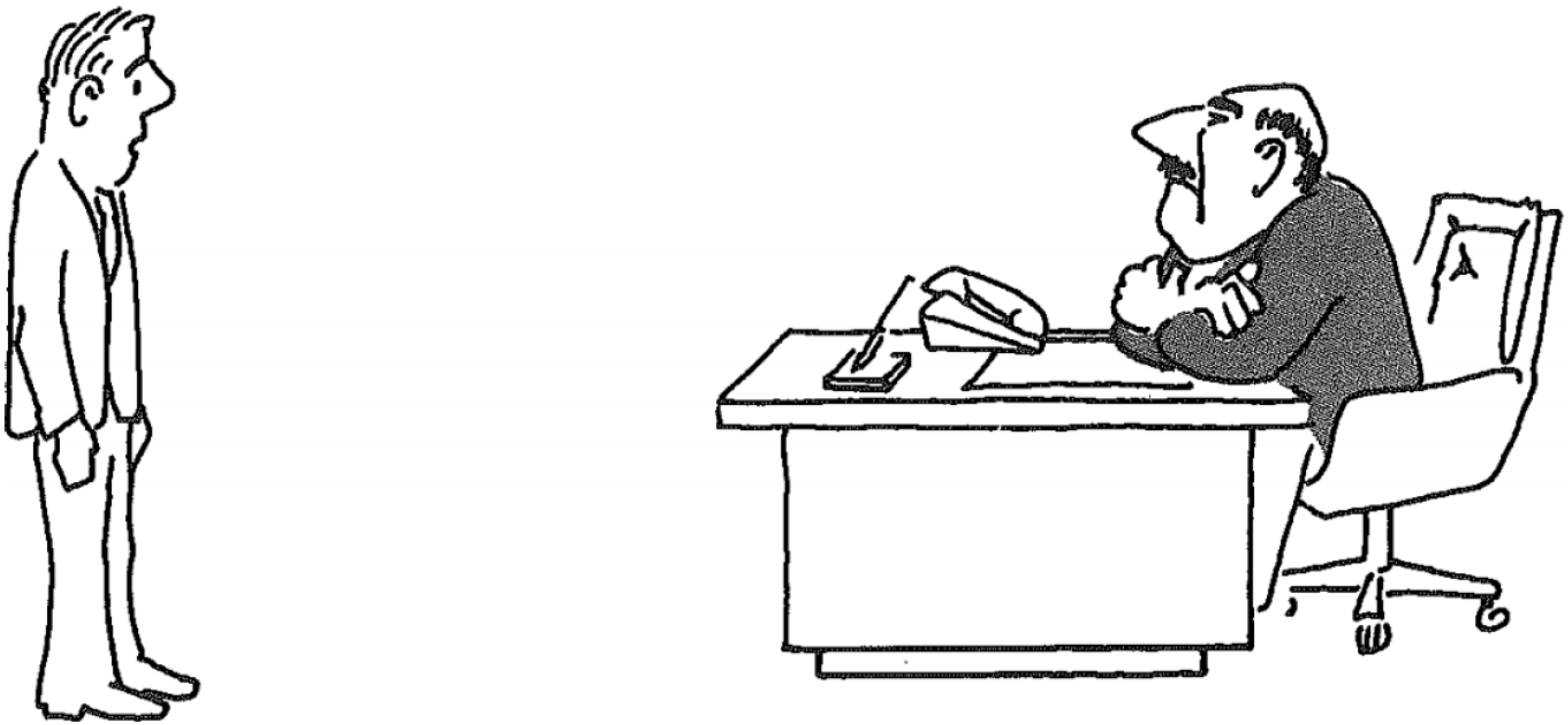
- Einführung: Modellieren auf Graphen ✓
- Teil 1: Graphen und Algorithmen auf Graphen
  - Graphen modellieren räumliche Zusammenhänge✓
  - Graphen modellieren Beziehungen✓
  - Algorithmen und (Problem-)Komplexität
    - Problemklassen P und NP
    - NP-vollständig und NP-schwer
    - Algorithmen für NP-schwere Probleme
- Teil 2: Lineare und ganzzahlige Optimierung auf Graphen

Literatur

Mehr zum Thema Komplexität als in Krumke & Noltemeier's 'Graphentheoretische Konzepte und Algorithmen' findet man zum Beispiel in

- Garey & Johnson: 'Computers and Intractability' (in Unibib vorhanden, aber leider nicht als E-Book)

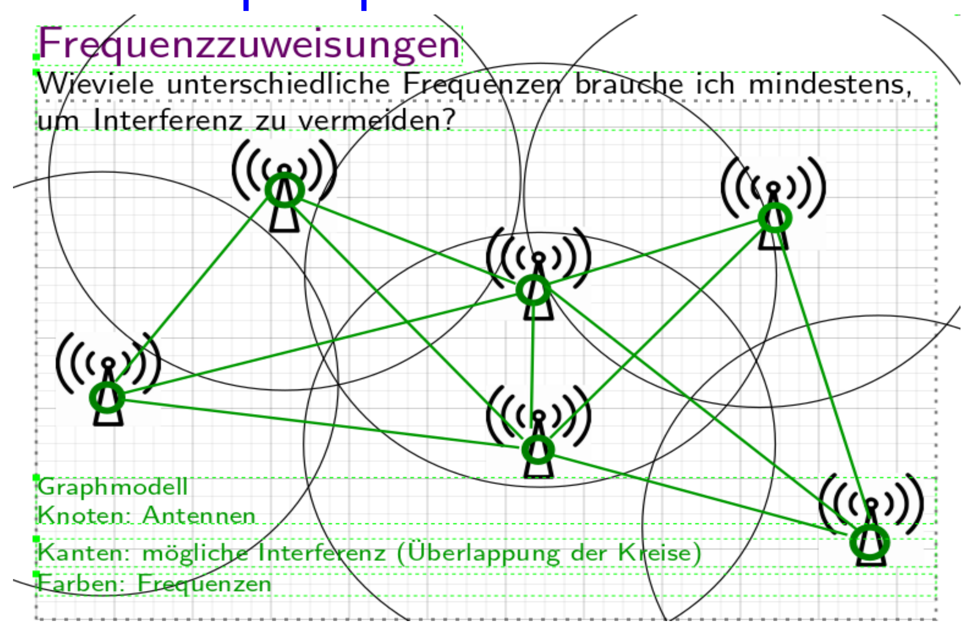
Unangenehme Situation…



"I can't find an efficient algorithm, I guess I'm just too dumb."

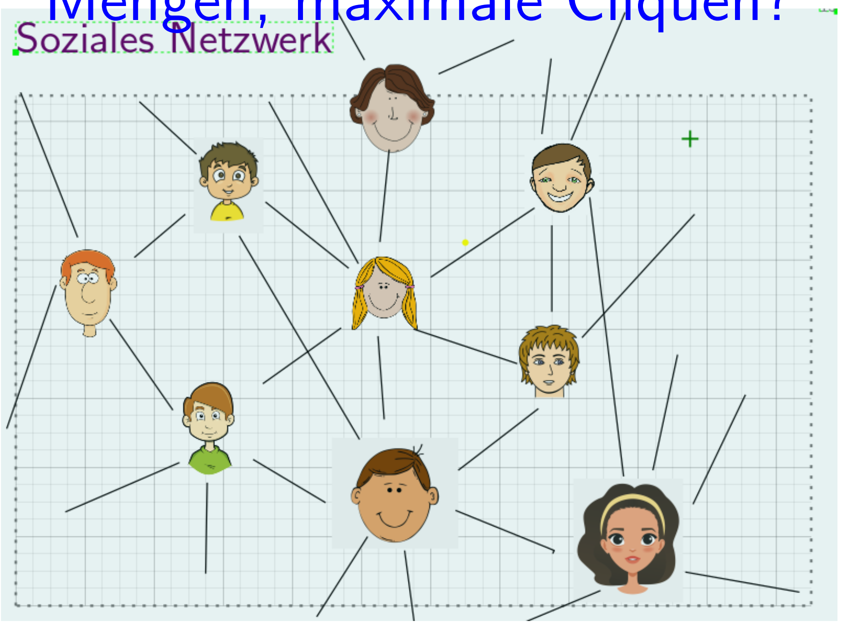
(Noch) offene Fragen...

Können wir mit einer Kombination aus Greedy-Algorithmus und Breitensuche jeden Graph optimal färben?

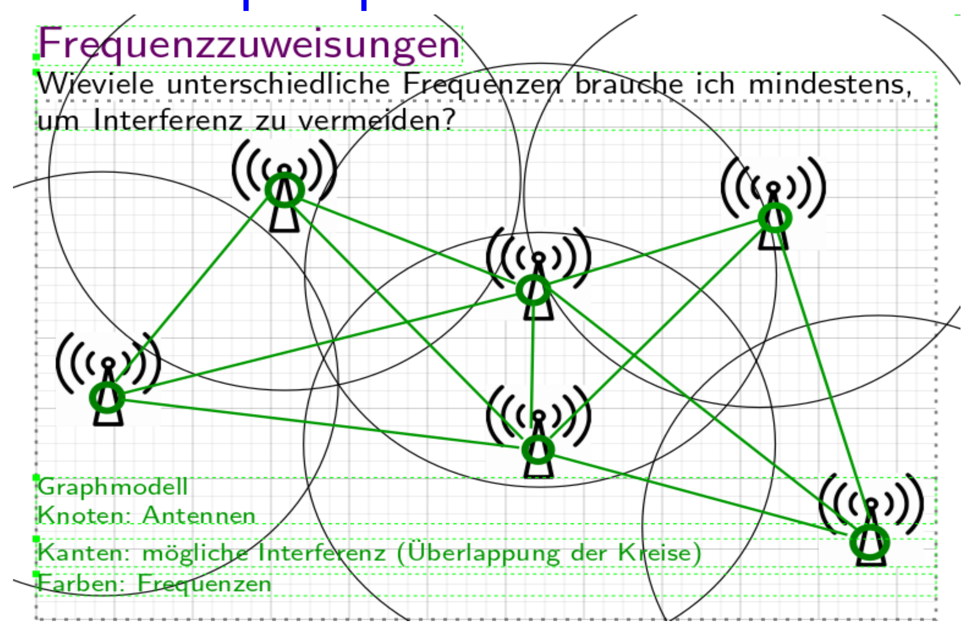


(Noch) offene Fragen...

Wie finden wir minimale Eckenüberdeckungen, maximale unabhängige Mengen, maximale Cliquen?

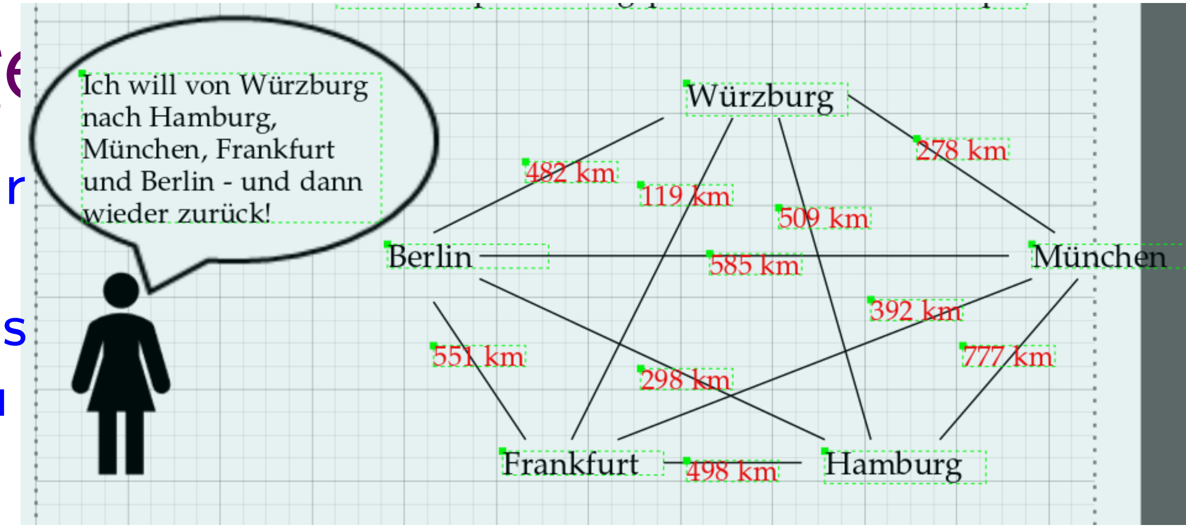


Können wir mit einer Kombination aus Greedy-Algorithmus und Breitensuche jeden Graph optimal färben?

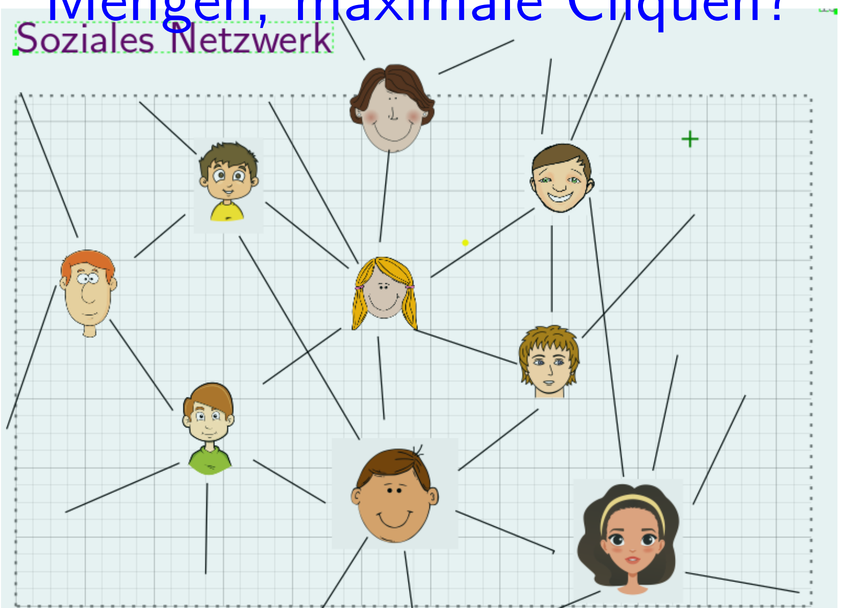


(Noch) offene Frage

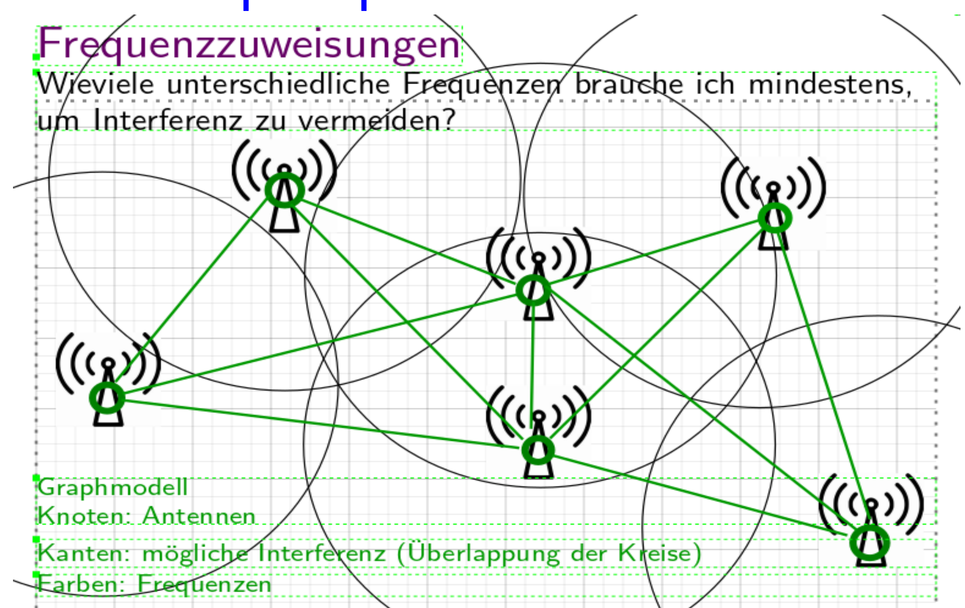
Wie finden wir eine optimale Tour für Handlungsreisende?
Und was ist ein guter Algorithmus um einen Hamiltonschen Kreis zu finden?



Wie finden wir minimale Eckenüberdeckungen, maximale unabhängige Mengen, maximale Cliquen?



Können wir mit einer Kombination aus Greedy-Algorithmus und Breitensuche jeden Graph optimal färben?



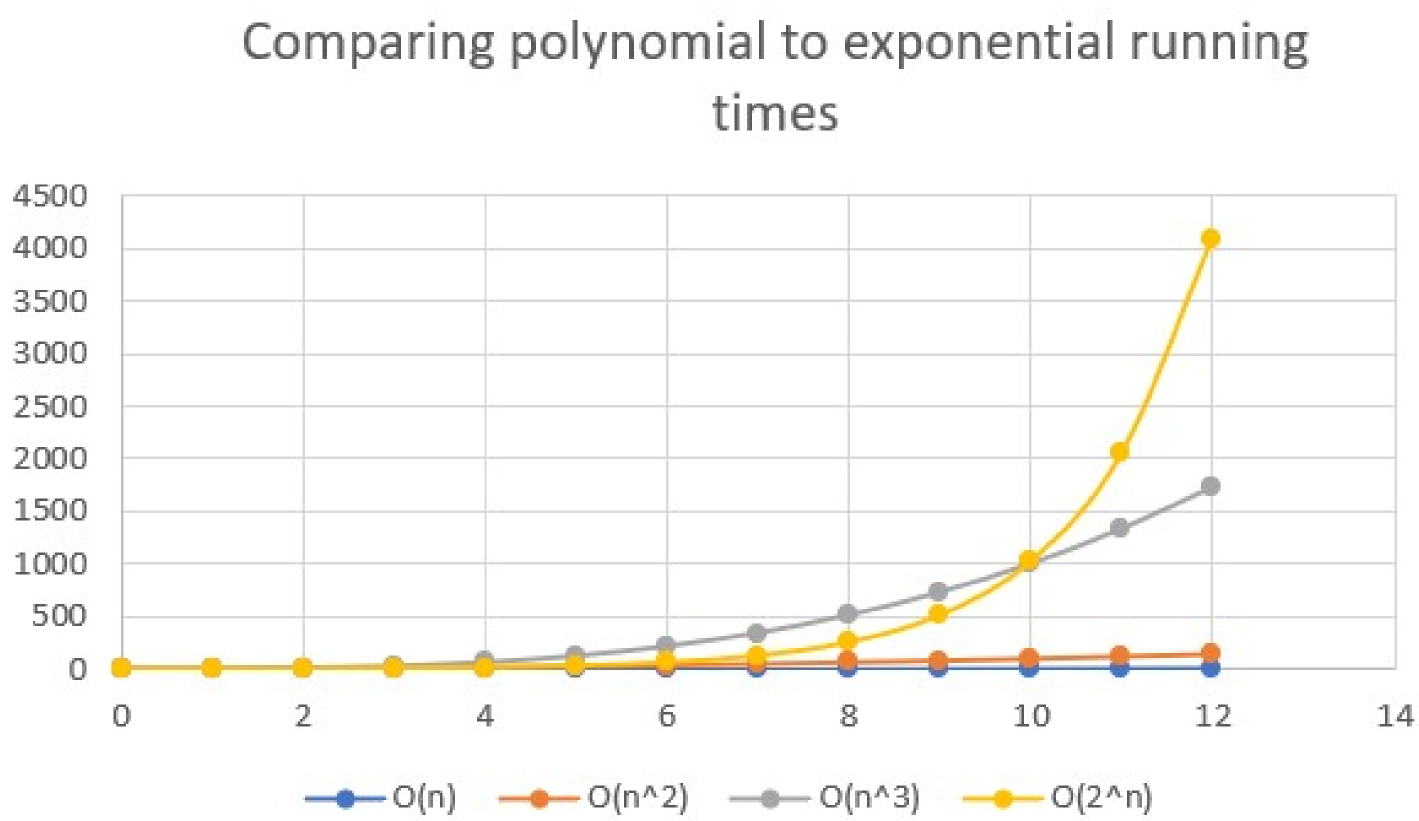
Gibt es *überhaupt* Algorithmen, die Optimallösungen für diese Probleme finden können?

Gibt es *überhaupt* Algorithmen, die Optimallösungen für diese Probleme finden können?

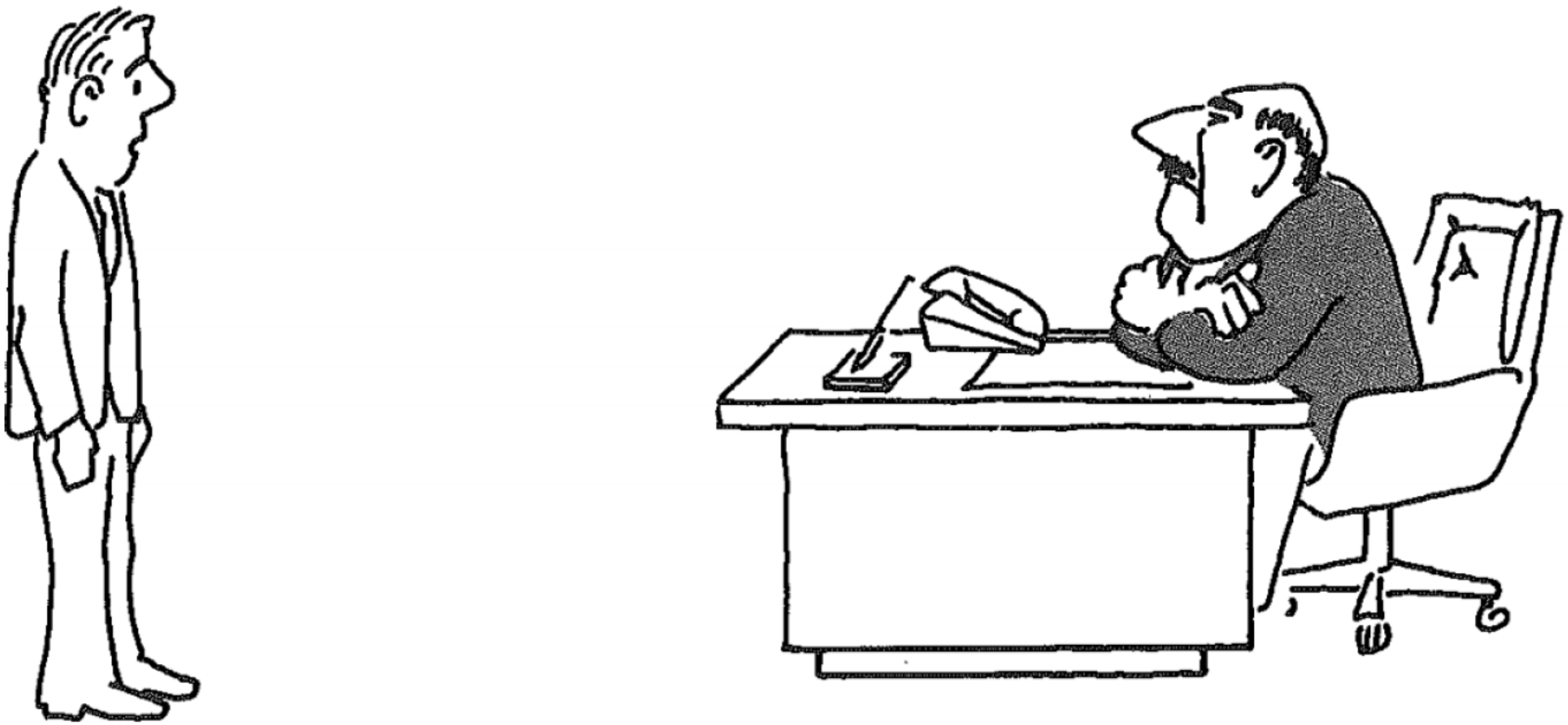
$\rightarrow$ bei den beschriebenen Problemen (und vielen anderen Problemen auf Graphen) ist es möglich, einfach alle möglichen Lösungen durchzuprobieren (Brute Force)
- das ist ein sehr ineffizienter, aber exakter Lösungsalgorithmus

Gibt es *überhaupt* Algorithmen, die Optimallösungen für diese Probleme finden können?

$\rightarrow$ bei den beschriebenen Problemen (und vielen anderen Problemen auf Graphen) ist es möglich, einfach alle möglichen Lösungen durchzuprobieren (Brute Force)
- das ist ein sehr ineffizienter, aber exakter Lösungsalgorithmus



Unangenehme Situation. . .

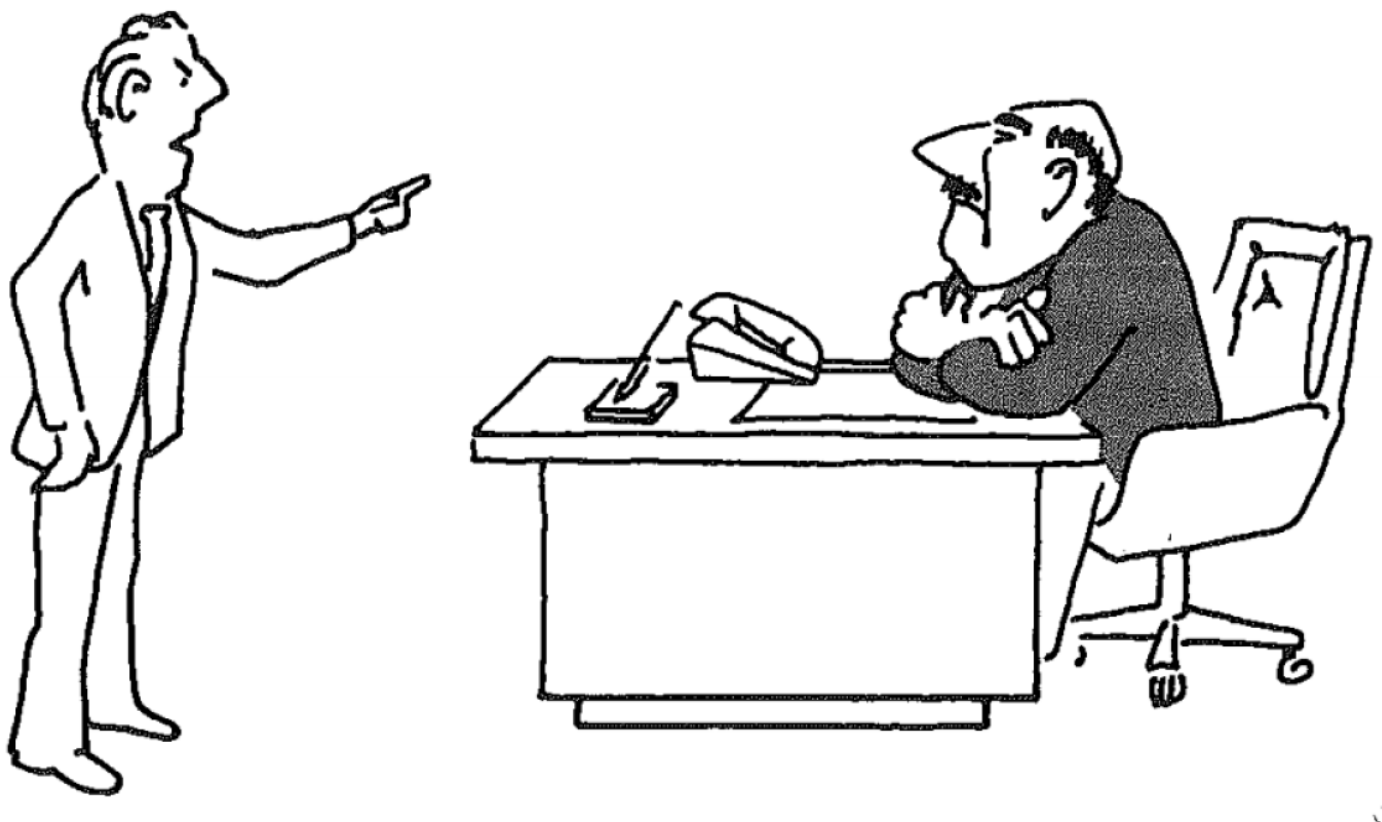


“I can’t find an efficient algorithm, I guess I’m just too dumb.”

. . . so wäre es angenehmer

"I can't find an efficient algorithm, because no such algorithm is possible!"

... so wäre es angenehmer



"I can't find an efficient algorithm, because no such algorithm is possible!"

Können wir beweisen, dass es für ein Problem keinen effizienten exakten Algorithmus gibt?

Entscheidungsprobleme

Gegeben: ...
Frage (JA/NEIN): ...

Beispiel:

Entscheidungsprobleme

Gegeben: ...
Frage (JA/NEIN): ...

Beispiel:

Entscheidungsproblem 'Eulerkreis?

Entscheidungsprobleme

Gegeben: . . .
Frage (JA/NEIN): . . .

Beispiel:

Entscheidungsproblem ’Eulerkreis?
Gegeben: Ein Graph G
Frage: Gibt es in Graph G einen Eulerkreis?

Entscheidungsprobleme

Gegeben: . . .
Frage (JA/NEIN): . . .

Beispiel:

Entscheidungsproblem 'Eulerkreis?
Gegeben: Ein Graph G
Frage: Gibt es in Graph G einen Eulerkreis?

Entscheidungsproblem 'Clique'

Entscheidungsprobleme

Gegeben: ...
Frage (JA/NEIN): ...

Beispiel:

Entscheidungsproblem 'Eulerkreis?
Gegeben: Ein Graph G
Frage: Gibt es in Graph G einen Eulerkreis?

Entscheidungsproblem 'Clique'
Gegeben: Ein Graph G, eine Zahl k
Frage: Gibt es in Graph G eine Clique von Größe k?

Problemklasse P

Ein (Entscheidungs- oder Optimierungs-)Problem liegt in **Problemklasse P**, wenn es einen Algorithmus gibt, der es in Polynomialzeit löst.

Problemklasse P

Ein (Entscheidungs- oder Optimierungs-)Problem liegt in **Problemklasse P**, wenn es einen Algorithmus gibt, der es in Polynomialzeit löst.

'es löst' heißt natürlich: der Algorithmus muss *jede Instanz* des Problems lösen können

Problemklasse P

Ein (Entscheidungs- oder Optimierungs-)Problem liegt in **Problemklasse P**, wenn es einen Algorithmus gibt, der es in Polynomialzeit löst.

'es löst' heißt natürlich: der Algorithmus muss *jede Instanz* des Problems lösen können

(Entscheidungs- /Optimierungs)Probleme in P?

Problemklasse P

Ein (Entscheidungs- oder Optimierungs-)Problem liegt in **Problemklasse P**, wenn es einen Algorithmus gibt, der es in Polynomialzeit löst.

'es löst' heißt natürlich: der Algorithmus muss *jede Instanz* des Problems lösen können

(Entscheidungs- /Optimierungs)Probleme in P?

Optimierungsprobleme: Kürzester Weg, Färben von Intervallgraphen, Färben von chordalen Graphen, Maximale Flüsse, Minimaler spannender Baum, ,...

Entscheidungsprobleme: Eulerkreis, Finden einer zulässigen Strömung, ...

Problemklasse P

Ein (Entscheidungs- oder Optimierungs-)Problem liegt in **Problemklasse P**, wenn es einen Algorithmus gibt, der es in Polynomialzeit löst.

'es löst' heißt natürlich: der Algorithmus muss *jede Instanz* des Problems lösen können

(Entscheidungs- /Optimierungs)Probleme in P?

Optimierungsprobleme: Kürzester Weg, Färben von Intervallgraphen, Färben von chordalen Graphen, Maximale Flüsse, Minimaler spannender Baum, ,. . .

Entscheidungsprobleme: Eulerkreis, Finden einer zulässigen Strömung, . . .

. . . und Entscheidungsversionen der oben genannten Optimierungsprobleme

Problemklasse P

Ein (Entscheidungs- oder Optimierungs-)Problem liegt in **Problemklasse P**, wenn es einen Algorithmus gibt, der es in Polynomialzeit löst.

'es löst' heißt natürlich: der Algorithmus muss *jede Instanz* des Problems lösen können

(Entscheidungs- /Optimierungs)Probleme in P?

Optimierungsprobleme: Kürzester Weg, Färben von Intervallgraphen, Färben von chordalen Graphen, Maximale Flüsse, Minimaler spannender Baum, ,...

Entscheidungsprobleme: Eulerkreis, Finden einer zulässigen Strömung, ...

...und Entscheidungsversionen der oben genannten Optimierungsprobleme

Ist ein Opt.problem in P, dann auch das zugehörige Entscheidungsproblem.

Problemklasse P

Ein (Entscheidungs- oder Optimierungs-)Problem liegt in **Problemklasse P**, wenn es einen Algorithmus gibt, der es in Polynomialzeit löst.

'es löst' heißt natürlich: der Algorithmus muss *jede Instanz* des Problems lösen können

(Entscheidungs- /Optimierungs)Probleme in P?

Optimierungsprobleme: Kürzester Weg, Färben von Intervallgraphen, Färben von chordalen Graphen, Maximale Flüsse, Minimaler spannender Baum, ,...

Entscheidungsprobleme: Eulerkreis, Finden einer zulässigen Strömung, ...

...und Entscheidungsversionen der oben genannten Optimierungsprobleme

Was ist denn das?

Ist ein Opt.problem in P, dann auch das zugehörige Entscheidungsproblem.

Optimierungsproblem OPT
Gegeben: Probleminstanz
Gesucht: Lösung im zulässiger Bereich, die Zielfunktion f minimiert / maximiert

Optimierungsproblem OPT
Gegeben: Probleminstanz
Gesucht: Lösung im zulässiger Bereich, die Zielfunktion f minimiert / maximiert

'zugehöriges' Entscheidungsproblem
Gegeben: Probleminstanz wie in OPT und Zahl k
Gesucht: Lösung X im zulässigen Bereich von k mit $f(X) \leq k$ (bzw $f(X) \geq k$, falls OPT Maximierungsproblem)

Optimierungsproblem OPT
Gegeben: Probleminstanz
Gesucht: Lösung im zulässiger Bereich, die Zielfunktion f minimiert / maximiert

'zugehöriges' Entscheidungsproblem
Gegeben: Probleminstanz wie in OPT und Zahl k
Gesucht: Lösung X im zulässigen Bereich von k mit $f(X) \leq k$ (bzw $f(X) \geq k$, falls OPT Maximierungsproblem)

Wenn ein Algorithmus A ein Optimierungsproblem OPT löst, dann lässt er sich auch zum Lösen des zugehörigen Entscheidungsproblems nutzen

Optimierungsproblem OPT
Gegeben: Probleminstanz
Gesucht: Lösung im zulässiger Bereich, die Zielfunktion f minimiert / maximiert

'zugehöriges' Entscheidungsproblem
Gegeben: Probleminstanz wie in OPT und Zahl k
Gesucht: Lösung X im zulässigen Bereich von k mit $f(X) \leq k$ (bzw $f(X) \geq k$, falls OPT Maximierungsproblem)

Wenn ein Algorithmus A ein Optimierungsproblem OPT löst, dann lässt er sich auch zum Lösen des zugehörigen Entscheidungsproblems nutzen

Ist OPT in Problemklasse P, dann ist auch das zugehörige Entscheidungsproblem in Problemklasse P.

Optimierungsproblem OPT
Gegeben: Probleminstanz
Gesucht: Lösung im zulässiger Bereich, die Zielfunktion f minimiert / maximiert

'zugehöriges' Entscheidungsproblem
Gegeben: Probleminstanz wie in OPT und Zahl k
Gesucht: Lösung X im zulässigen Bereich von k mit $f(X) \leq k$ (bzw $f(X) \geq k$, falls OPT Maximierungsproblem)

Wenn ein Algorithmus A ein Optimierungsproblem OPT löst, dann lässt er sich auch zum Lösen des zugehörigen Entscheidungsproblems nutzen

Ist OPT in Problemklasse P, dann ist auch das zugehörige Entscheidungsproblem in Problemklasse P.

Gilt das auch andersherum? (Übungsblatt)

Problemklasse NP

Und was ist mit dem Handlungsreisendenproblem, Färbungsproblem, Entscheidungsproblem Clique usw?

Problemklasse NP

Und was ist mit dem Handlungsreisendenproblem, Färbungsproblem, Entscheidungsproblem Clique usw?

Ein *Entscheidungs*problem liegt in **Problemklasse NP**, wenn jede JA-Instanz ein **Zertifikat** polynomieller Länge besitzt, das in polynomieller Zeit verifiziert werden kann.

Problemklasse NP

Und was ist mit dem Handlungsreisendenproblem, Färbungsproblem, Entscheidungsproblem Clique usw?

gemeint ist: polynomiell in der Länge der Eingabe der Instanz $|I|$

Ein *Entscheidungs*problem liegt in **Problemklasse NP**, wenn jede JA-Instanz ein **Zertifikat** polynomieller Länge besitzt, das in polynomieller Zeit verifiziert werden kann.

Problemklasse NP

Und was ist mit dem Handlungsreisendenproblem, Färbungsproblem, Entscheidungsproblem Clique usw?

gemeint ist: polynomiell in der Länge der Eingabe der Instanz $|I|$

Ein *Entscheidungs*problem liegt in **Problemklasse NP**, wenn jede JA-Instanz ein **Zertifikat** polynomieller Länge besitzt, das in polynomieller Zeit verifiziert werden kann.

Entscheidungsproblem 'Clique'
Gegeben: Ein Graph G, eine Zahl k
Frage: Gibt es in Graph G eine Clique von Größe k

Problemklasse NP

Und was ist mit dem Handlungsreisendenproblem, Färbungsproblem, Entscheidungsproblem Clique usw?

gemeint ist: polynomiell in der Länge der Eingabe der Instanz $|I|$

Ein *Entscheidungs*problem liegt in **Problemklasse NP**, wenn jede JA-Instanz ein **Zertifikat** polynomieller Länge besitzt, das in polynomieller Zeit verifiziert werden kann.

Entscheidungsproblem 'Clique'
Gegeben: Ein Graph G, eine Zahl k
Frage: Gibt es in Graph G eine Clique von Größe k

Zertifikat für JA-Instanz?

Problemklasse NP

Und was ist mit dem Handlungsreisendenproblem, Färbungsproblem, Entscheidungsproblem Clique usw?

gemeint ist: polynomiell in der Länge der Eingabe der Instanz $|I|$

Ein *Entscheidungs*problem liegt in **Problemklasse NP**, wenn jede JA-Instanz ein **Zertifikat** polynomieller Länge besitzt, das in polynomieller Zeit verifiziert werden kann.

Entscheidungsproblem 'Clique'
Gegeben: Ein Graph G, eine Zahl k
Frage: Gibt es in Graph G eine Clique von Größe k

Zertifikat für JA-Instanz?
Eine Knotenmenge $C \subset V$ mit k Knoten, die alle untereinander verbunden sind (aka: eine Clique der Größe k)

Problemklasse NP

Und was ist mit dem Handlungsreisendenproblem, Färbungsproblem, Entscheidungsproblem Clique usw?

gemeint ist: polynomiell in der Länge der Eingabe der Instanz $|I|$

Ein *Entscheidungs*problem liegt in **Problemklasse NP**, wenn jede JA-Instanz ein **Zertifikat** polynomieller Länge besitzt, das in polynomieller Zeit verifiziert werden kann.

Entscheidungsproblem 'Clique'
Gegeben: Ein Graph G, eine Zahl k
Frage: Gibt es in Graph G eine Clique von Größe k

Zertifikat für JA-Instanz?
Eine Knotenmenge $C \subset V$ mit k Knoten, die alle untereinander verbunden sind (aka: eine Clique der Größe k)

Wie in Polynomialzeit überprüfen?

Problemklasse NP

Und was ist mit dem Handlungsreisendenproblem, Färbungsproblem, Entscheidungsproblem Clique usw?

gemeint ist: polynomiell in der Länge der Eingabe der Instanz $|I|$

Ein *Entscheidungs*problem liegt in **Problemklasse NP**, wenn jede JA-Instanz ein **Zertifikat** polynomieller Länge besitzt, das in polynomieller Zeit verifiziert werden kann.

Entscheidungsproblem 'Clique'
Gegeben: Ein Graph G, eine Zahl k
Frage: Gibt es in Graph G eine Clique von Größe k

Zertifikat für JA-Instanz?
Eine Knotenmenge $C \subset V$ mit k Knoten, die alle untereinander verbunden sind (aka: eine Clique der Größe k)

Wie in Polynomialzeit überprüfen?
Zähle: Ist $|C| \geq k$.
Für jedes Knotenpaar $\{u, v\} \in C \times C$ mit $u \neq v$, überprüfe, ob $\{u, v\} \in E$

Problemklasse NP

Und was ist mit dem Handlungsreisendenproblem, Färbungsproblem, Entscheidungsproblem Clique usw?

gemeint ist: polynomiell in der Länge der Eingabe der Instanz $|I|$

Ein *Entscheidungs*problem liegt in **Problemklasse NP**, wenn jede JA-Instanz ein **Zertifikat** polynomieller Länge besitzt, das in polynomieller Zeit verifiziert werden kann.

Entscheidungsproblem 'Clique'
Gegeben: Ein Graph G, eine Zahl k
Frage: Gibt es in Graph G eine Clique von Größe k

Zertifikat für JA-Instanz?
Eine Knotenmenge $C \subset V$ mit k Knoten, die alle untereinander verbunden sind (aka: eine Clique der Größe k)

Wie in Polynomialzeit überprüfen?
Zähle: Ist $|C| \geq k$.
Für jedes Knotenpaar $\{u, v\} \in C \times C$ mit $u \neq v$, überprüfe, ob $\{u, v\} \in E$
$\rightarrow O(|V|^2)$

Problemklasse NP

Und was ist mit dem Handlungsreisendenproblem, Färbungsproblem, Entscheidungsproblem Clique usw?

gemeint ist: polynomiell in der Länge der Eingabe der Instanz $|I|$

Ein *Entscheidungs*problem liegt in **Problemklasse NP**, wenn jede JA-Instanz ein **Zertifikat** polynomieller Länge besitzt, das in polynomieller Zeit verifiziert werden kann.

Entscheidungsproblem 'Eulerkreis?
Gegeben: Ein Graph G
Frage: Gibt es in Graph G einen Eulerkreis?

Zertifikat für JA-Instanz?

Problemklasse NP

Und was ist mit dem Handlungsreisendenproblem, Färbungsproblem, Entscheidungsproblem Clique usw?

gemeint ist: polynomiell in der Länge der Eingabe der Instanz $|I|$

Ein *Entscheidungs*problem liegt in **Problemklasse NP**, wenn jede JA-Instanz ein **Zertifikat** polynomieller Länge besitzt, das in polynomieller Zeit verifiziert werden kann.

Entscheidungsproblem 'Eulerkreis?
Gegeben: Ein Graph G
Frage: Gibt es in Graph G einen Eulerkreis?

Zertifikat für JA-Instanz?

Eine Knotenfolge in G, die einen Eulerkreis darstellt

Problemklasse NP

Und was ist mit dem Handlungsreisendenproblem, Färbungsproblem, Entscheidungsproblem Clique usw?

gemeint ist: polynomiell in der Länge der Eingabe der Instanz $|I|$

Ein *Entscheidungs*problem liegt in **Problemklasse NP**, wenn jede JA-Instanz ein **Zertifikat** polynomieller Länge besitzt, das in polynomieller Zeit verifiziert werden kann.

Entscheidungsproblem 'Eulerkreis?
Gegeben: Ein Graph G
Frage: Gibt es in Graph G einen Eulerkreis?

Zertifikat für JA-Instanz?

Eine Knotenfolge in G, die einen Eulerkreis darstellt

Wie in Polynomialzeit überprüfen?

Problemklasse NP

Und was ist mit dem Handlungsreisendenproblem, Färbungsproblem, Entscheidungsproblem Clique usw?

gemeint ist: polynomiell in der Länge der Eingabe der Instanz $|I|$

Ein *Entscheidungs*problem liegt in **Problemklasse NP**, wenn jede JA-Instanz ein **Zertifikat** polynomieller Länge besitzt, das in polynomieller Zeit verifiziert werden kann.

Entscheidungsproblem 'Eulerkreis?
Gegeben: Ein Graph G
Frage: Gibt es in Graph G einen Eulerkreis?

Zertifikat für JA-Instanz?

Eine Knotenfolge in G, die einen Eulerkreis darstellt

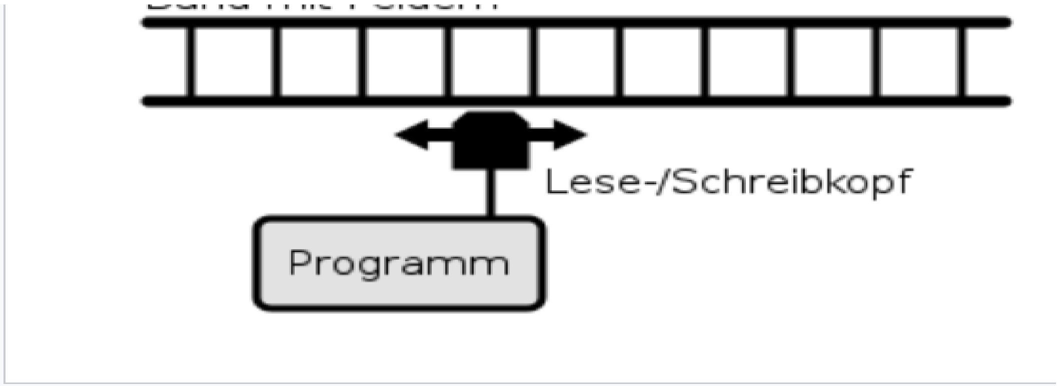
Wie in Polynomialzeit überprüfen?

Knotenfolge entlang laufen: sind zwei aufeinander folgende Knoten immer durch Kante verbunden?
Sind alle Kanten benutzt?

'Ursprüngliche' Definitionen von P und NP
(nur der Vollständigkeit halber)

Ein Entscheidungsproblem liegt in **Problemklasse P**, wenn

- **es einen Algorithmus gibt, der es in Polynomialzeit löst.**
- oder alternativ: wenn es auf einer deterministischen Turingmaschine in Polynomialzeit gelöst werden kann.



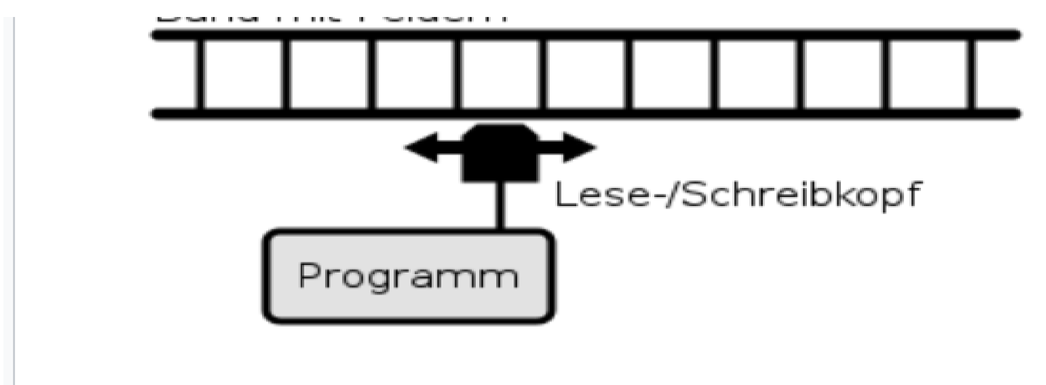
'Ursprüngliche' Definitionen von P und NP
(nur der Vollständigkeit halber)

Ein Entscheidungsproblem liegt in **Problemklasse P**, wenn

- **es einen Algorithmus gibt, der es in Polynomialzeit löst.**
- oder alternativ: wenn es auf einer deterministischen Turingmaschine in Polynomialzeit gelöst werden kann.

Ein Entscheidungsproblem liegt in **Problemklasse NP**, wenn

- **jede JA-Instanz ein Zertifikat polynomieller Länge besitzt, das in polynomieller Zeit verifiziert werden kann.**
- das Problem auf einer *nichtdeterministischen* Turingmaschine in polynomieller Zeit gelöst werden kann.



'Ursprüngliche' Definitionen von P und NP
(nur der Vollständigkeit halber)

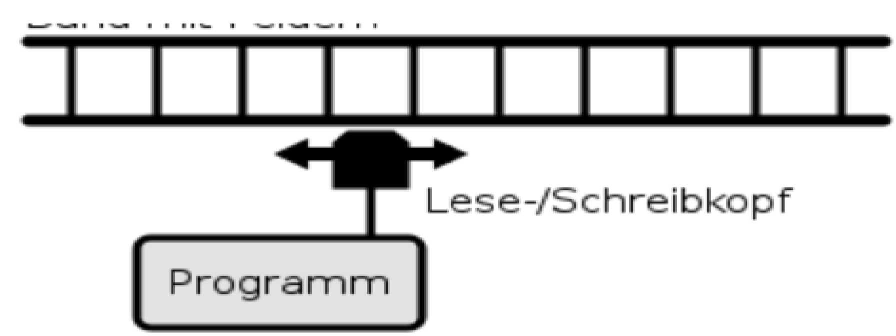
Ein Entscheidungsproblem liegt in **Problemklasse P**, wenn

- **es einen Algorithmus gibt, der es in Polynomialzeit löst.**
- oder alternativ: wenn es auf einer deterministischen Turingmaschine in Polynomialzeit gelöst werden kann.

Ein Entscheidungsproblem liegt in **Problemklasse NP**, wenn

- **jede JA-Instanz ein Zertifikat polynomieller Länge besitzt, das in polynomieller Zeit verifiziert werden kann.**
- das Problem auf einer *nichtdeterministischen* Turingmaschine in polynomieller Zeit gelöst werden kann.

Was ist mit Optimierungsproblemen?



P und NP

Wie hängen die Problemklassen P und NP zusammen?

P und NP

Wie hängen die Problemklassen P und NP zusammen?

Jedes Entscheidungsproblem, das in P ist, ist auch in NP.

P und NP

Wie hängen die Problemklassen P und NP zusammen?

Jedes Entscheidungsproblem, das in P ist, ist auch in NP.

Was ist ein geeignetes Zertifikat für ein Entscheidungsproblem aus P?

P und NP

Wie hängen die Problemklassen P und NP zusammen?

Jedes Entscheidungsproblem, das in P ist, ist auch in NP.

Was ist ein geeignetes Zertifikat für ein Entscheidungsproblem aus P?

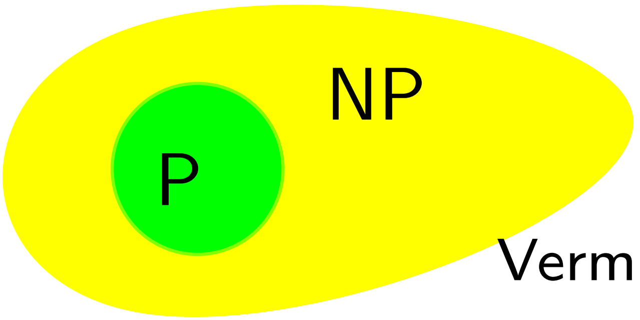
Gibt es Probleme, die in NP sind, aber nicht in P?

P und NP

Wie hängen die Problemklassen P und NP zusammen?

Jedes Entscheidungsproblem, das in P ist, ist auch in NP.

Was ist ein geeignetes Zertifikat für ein Entscheidungsproblem aus P?



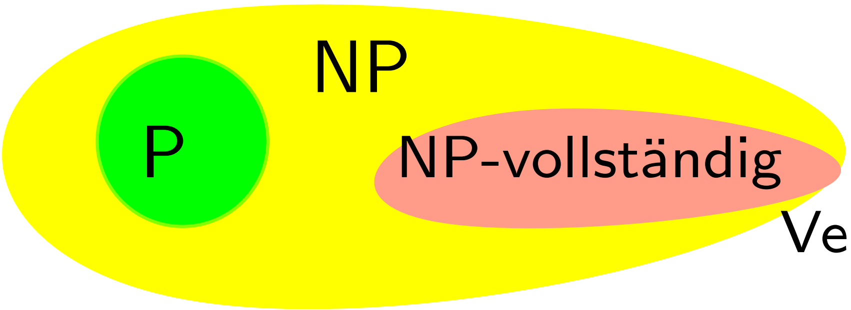
Vermutung: $\mathrm{P} \neq \mathrm{NP}$

Gibt es Probleme, die in NP sind, aber nicht in P?

Die meisten Leute vermuten: Ja!
Aber beweisen konnte es noch keiner.

NP-vollständig

Die Problemklasse der *am schwersten zu lösenden* Entscheidungsprobleme nennen wir **NP-vollständige** Probleme.



Vermutung: P $\neq$ NP

NP-vollständig

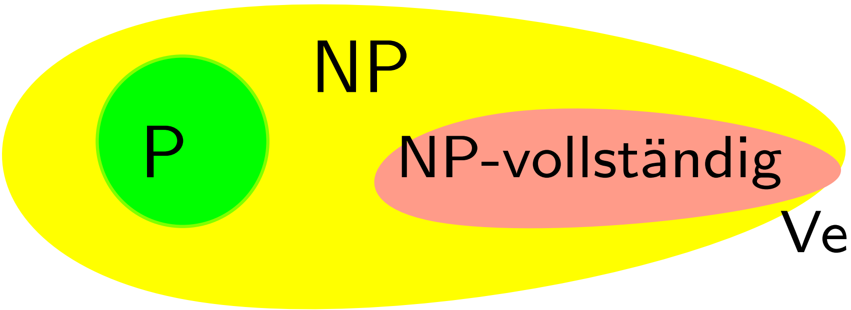
Die Problemklasse der *am schwersten zu lösenden* Entscheidungsprobleme nennen wir **NP-vollständige** Probleme.

Ein Entscheidungsproblem heißt NP-vollständig, wenn

1. es in NP liegt und
2. wenn ein polynomieller Lösungsalgorithmus für dieses Problem uns ermöglichen würde, *jedes* andere Problem in NP in Polynomialzeit zu lösen.

NP-vollständig

Die Problemklasse der *am schwersten zu lösenden* Entscheidungsprobleme nennen wir **NP-vollständige** Probleme.



Vermutung: P $\neq$ NP

Ein Entscheidungsproblem heißt NP-vollständig, wenn

1. es in NP liegt und
2. wenn ein polynomieller Lösungsalgorithmus für dieses Problem uns ermöglichen würde, *jedes* andere Problem in NP in Polynomialzeit zu lösen.

Wie soll das denn gehen???

Polynomialzeitreduktionen

Seien X und Y zwei Entscheidungsprobleme.
Wir sagen, dass X (in Polynomialzeit) **reduzierbar** auf Y ist, wenn wir jede Probleminstanz I von X (in Polynomialzeit) in eine Probleminstanz $f(I)$ von Y umformen können, so dass gilt

I ist eine JA-Instanz von Problem X

$\Leftrightarrow$

$f(I)$ ist eine JA-Instanz von Problem Y

Polynomialzeitreduktionen

Seien X und Y zwei Entscheidungsprobleme.
Wir sagen, dass X (in Polynomialzeit) **reduzierbar** auf Y ist, wenn wir jede Probleminstanz I von X (in Polynomialzeit) in eine Probleminstanz $f(I)$ von Y umformen können, so dass gilt

$$I \text{ ist eine JA-Instanz von Problem X}$$
$$\Leftrightarrow$$
$$f(I) \text{ ist eine JA-Instanz von Problem Y}$$

Wir schreiben dann $X \leq_p Y$.

Polynomialzeitreduktionen

Seien X und Y zwei Entscheidungsprobleme.
Wir sagen, dass X (in Polynomialzeit) **reduzierbar** auf Y ist, wenn wir jede Probleminstanz I von X (in Polynomialzeit) in eine Probleminstanz $f(I)$ von Y umformen können, so dass gilt

$$I \text{ ist eine JA-Instanz von Problem X}$$
$$\Leftrightarrow$$
$$f(I) \text{ ist eine JA-Instanz von Problem Y}$$

Wir schreiben dann $X \leq_p Y$.

Wir nennen f (polynomielle) **Transformation**.

Polynomialzeitreduktionen

Seien X und Y zwei Entscheidungsprobleme.
Wir sagen, dass X (in Polynomialzeit) **reduzierbar** auf Y ist, wenn wir jede Probleminstanz I von X (in Polynomialzeit) in eine Probleminstanz $f(I)$ von Y umformen können, so dass gilt

$$I \text{ ist eine JA-Instanz von Problem X}$$
$$\Leftrightarrow$$
$$f(I) \text{ ist eine JA-Instanz von Problem Y}$$

Wir schreiben dann $X \leq_p Y$.

Wir nennen f (polynomielle) **Transformation**.

Intuition: $X \leq_p Y$ heißt: X ist 'höchstens so schwer' wie Y.

Polynomialzeitreduktionen

Seien X und Y zwei Entscheidungsprobleme.
Wir sagen, dass X (in Polynomialzeit) **reduzierbar** auf Y ist, wenn wir jede Probleminstanz I von X (in Polynomialzeit) in eine Probleminstanz $f(I)$ von Y umformen können, so dass gilt

$$I \text{ ist eine JA-Instanz von Problem X} \\ \Leftrightarrow \\ f(I) \text{ ist eine JA-Instanz von Problem Y}$$

Wir schreiben dann $X \leq_p Y$.

Wir nennen f (polynomielle) **Transformation**.

Intuition: $X \leq_p Y$ heißt: X ist 'höchstens so schwer' wie Y.

Konsequenz:

- Können wir Y in Polynomialzeit lösen, dann auch X.

Polynomialzeitreduktionen

Seien X und Y zwei Entscheidungsprobleme.
Wir sagen, dass X (in Polynomialzeit) **reduzierbar** auf Y ist, wenn wir jede Probleminstanz I von X (in Polynomialzeit) in eine Probleminstanz $f(I)$ von Y umformen können, so dass gilt

I ist eine JA-Instanz von Problem X

$\Leftrightarrow$

$f(I)$ ist eine JA-Instanz von Problem Y

Wir schreiben dann $X \leq_p Y$.

Wir nennen f (polynomielle) **Transformation**.

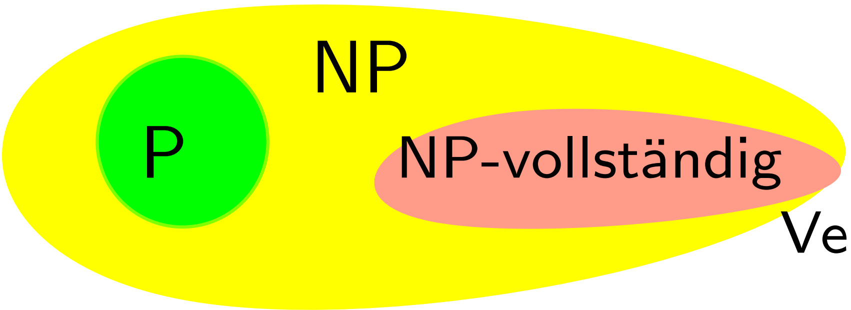
Intuition: $X \leq_p Y$ heißt: X ist 'höchstens so schwer' wie Y.

Konsequenz:

- Können wir Y in Polynomialzeit lösen, dann auch X.
- Können wir X nicht in Polynomialzeit lösen, dann können wir auch Y nicht in Polynomialzeit lösen.

NP-vollständig

Die Problemklasse der *am schwersten zu lösenden* Entscheidungsprobleme nennen wir **NP-vollständige** Probleme.



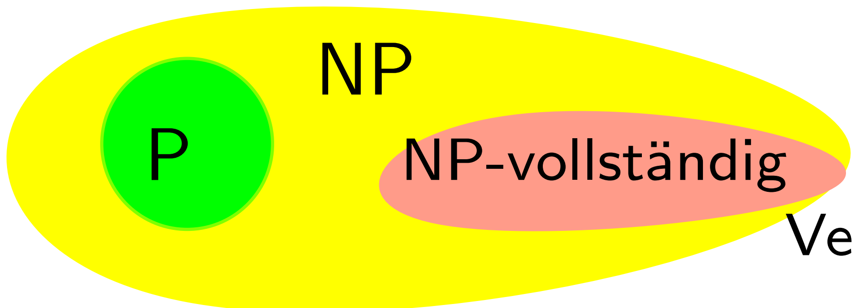
Vermutung: P $\neq$ NP

Formal: Ein Entscheidungsproblem X heißt NP-vollständig, wenn

1. es in NP liegt und
2. wenn ein polynomieller Lösungsalgorithmus für dieses Problem uns ermöglichen würde, *jedes* andere Problem in NP in Polynomialzeit zu lösen.

NP-vollständig

Die Problemklasse der *am schwersten zu lösenden* Entscheidungsprobleme nennen wir **NP-vollständige** Probleme.



Vermutung: $P \neq NP$

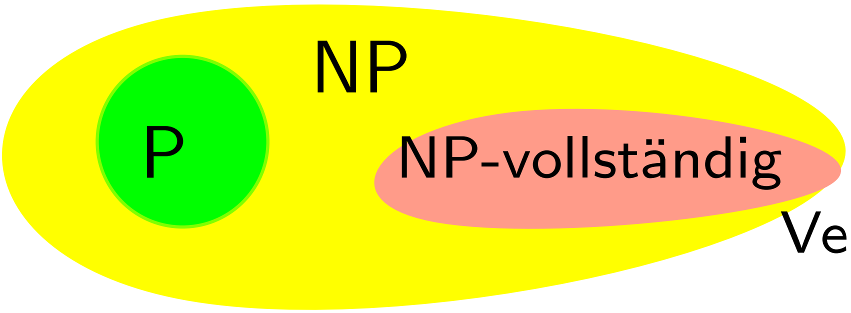
Formal: Ein Entscheidungsproblem X heißt NP-vollständig, wenn

1. es in NP liegt und
2. wenn ein polynomieller Lösungsalgorithmus für dieses Problem uns ermöglichen würde, *jedes* andere Problem in NP in Polynomialzeit zu lösen.

   Das heißt: für jedes Problem Y in NP gibt es eine Reduktion in Polynomialzeit auf X: $Y \leq_p X \; \forall Y \in \mathrm{NP}$

NP-vollständig

Die Problemklasse der *am schwersten zu lösenden* Entscheidungsprobleme nennen wir **NP-vollständige** Probleme.



Vermutung: P $\neq$ NP

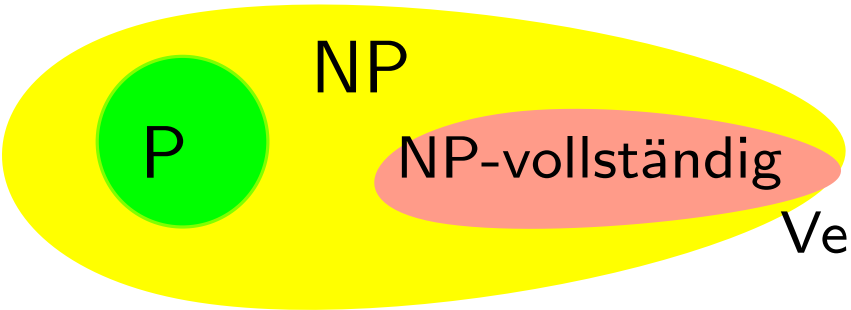
Formal: Ein Entscheidungsproblem X heißt NP-vollständig, wenn

1. es in NP liegt und
2. wenn ein polynomieller Lösungsalgorithmus für dieses Problem uns ermöglichen würde, *jedes* andere Problem in NP in Polynomialzeit zu lösen.
   Das heißt: für jedes Problem Y in NP gibt es eine Reduktion in Polynomialzeit auf X: $Y \leq_p X \ \forall Y \in \mathrm{NP}$

Wie soll das denn gehen???

NP-vollständig

Die Problemklasse der *am schwersten zu lösenden* Entscheidungsprobleme nennen wir **NP-vollständige** Probleme.



Vermutung: $P \neq NP$

Formal: Ein Entscheidungsproblem X heißt NP-vollständig, wenn

1. es in NP liegt und
2. wenn ein polynomieller Lösungsalgorithmus für dieses Problem uns ermöglichen würde, *jedes* andere Problem in NP in Polynomialzeit zu lösen.

Es reicht zu zeigen: für *ein NP-vollständiges* Problem Y gibt es eine Polynomialzeitreduktion auf X

$\leq_p$ ist transitiv

Lemma

Seien X, Y, Z drei Entscheidungsprobleme.
Wenn $X \leq_p Y$ und $Y \leq_p Z$, dann gilt auch $X \leq_p Z$.

$\leq_p$ ist transitiv

Lemma

Seien X, Y, Z drei Entscheidungsprobleme.
Wenn $X \leq_p Y$ und $Y \leq_p Z$, dann gilt auch $X \leq_p Z$.

Beweis: Beide Transformationen nacheinander ausführen.

$\leq_p$ ist transitiv

Lemma
Seien X, Y, Z drei Entscheidungsprobleme.
Wenn $X \leq_p Y$ und $Y \leq_p Z$, dann gilt auch $X \leq_p Z$.

Beweis: Beide Transformationen nacheinander ausführen.

Konsequenz: Um zu zeigen, dass ein Problem X NP-vollständig ist, reicht es, zu zeigen, dass *ein* anderes NP-vollständiges Problem darauf reduziert werden kann.

Beispiel Polynomialzeitreduktion

Satz: Entscheidungsproblem Hamiltonscher Kreis (HC) $\leq_p$ Entscheidungsproblem Handlungsreisendenproblem (TSP)

Beispiel Polynomialzeitreduktion

Satz: Entscheidungsproblem Hamiltonscher Kreis (HC) $\leq_p$ Entscheidungsproblem Handlungsreisendenproblem (TSP)

Entscheidungsproblem 'Hamiltonscher Kreis' (HC)?
Gegeben: Ein Graph G
Frage: Gibt es in Graph G einen hamiltonschen Kreis?

Beispiel Polynomialzeitreduktion

Satz: Entscheidungsproblem Hamiltonscher Kreis (HC) $\leq_p$ Entscheidungsproblem Handlungsreisendenproblem (TSP)

Entscheidungsproblem 'Hamiltonscher Kreis' (HC)?
Gegeben: Ein Graph G
Frage: Gibt es in Graph G einen hamiltonschen Kreis?

Optimierungsproblem TSP:
Gegeben: Vollständiger Graph $G = (V, E)$, Kantenlabels l_e für $e \in E$
Gesucht: Ein Kreis $K = (V_K, E_K)$, der alle Knoten besucht, so dass $\sum_{e \in E_K} l_e$ minimal

Beispiel Polynomialzeitreduktion

Satz: Entscheidungsproblem Hamiltonscher Kreis (HC) $\leq_p$ Entscheidungsproblem Handlungsreisendenproblem (TSP)

Entscheidungsproblem 'Hamiltonscher Kreis' (HC)?
Gegeben: Ein Graph G
Frage: Gibt es in Graph G einen hamiltonschen Kreis?

Entscheidungsproblem TSP:
Gegeben: Vollständiger Graph $G = (V, E)$, Kantenlabels l_e für $e \in E$, Zahl k
Gesucht: Ein Kreis $K = (V_K, E_K)$, der alle Knoten besucht, so dass $\sum_{e \in E_K} l_e \leq k$

Beispiel Polynomialzeitreduktion

Satz: Entscheidungsproblem Hamiltonscher Kreis (HC) $\leq_p$ Entscheidungsproblem Handlungsreisendenproblem (TSP)

Beweis:

Was müssen wir tun, um das zu zeigen?

Beispiel Polynomialzeitreduktion

Satz: Entscheidungsproblem Hamiltonscher Kreis (HC) $\leq_p$ Entscheidungsproblem Handlungsreisendenproblem (TSP)

Beweis:

Was müssen wir tun, um das zu zeigen?

Wir suchen eine Polynomialzeitreduktion f, die jede Instanz I von HC in eine Instanz $f(I)$ von TSP überführt,
so dass JA-Instanzen von HC auf JA-Instanzen von TSP abgebildet werden, und NEIN-Instanzen von HC auf NEIN-Instanzen von TSP.

Beispiel Polynomialzeitreduktion

Satz: Entscheidungsproblem Hamiltonscher Kreis (HC) $\leq_p$ Entscheidungsproblem Handlungsreisendenproblem (TSP)

Beweis:

Was müssen wir tun, um das zu zeigen?

Wir suchen eine Polynomialzeitreduktion f, die jede Instanz I von HC in eine Instanz $f(I)$ von TSP überführt,
so dass JA-Instanzen von HC auf JA-Instanzen von TSP abgebildet werden, und NEIN-Instanzen von HC auf NEIN-Instanzen von TSP.

Sei $G = (V, E)$ eine Instanz von HC.

Beispiel Polynomialzeitreduktion

Satz: Entscheidungsproblem Hamiltonscher Kreis (HC) $\leq_p$ Entscheidungsproblem Handlungsreisendenproblem (TSP)

Beweis:

Was müssen wir tun, um das zu zeigen?

Wir suchen eine Polynomialzeitreduktion f, die jede Instanz I von HC in eine Instanz $f(I)$ von TSP überführt,
so dass JA-Instanzen von HC auf JA-Instanzen von TSP abgebildet werden, und NEIN-Instanzen von HC auf NEIN-Instanzen von TSP.

Sei $G = (V, E)$ eine Instanz von HC.

Wir konstruieren eine Instanz $f(G) = (\tilde{G}, k)$ von TSP wie folgt:

Beispiel Polynomialzeitreduktion

Satz: Entscheidungsproblem Hamiltonscher Kreis (HC) $\leq_p$ Entscheidungsproblem Handlungsreisendenproblem (TSP)

Beweis:

Was müssen wir tun, um das zu zeigen?

Wir suchen eine Polynomialzeitreduktion f, die jede Instanz I von HC in eine Instanz $f(I)$ von TSP überführt,
so dass JA-Instanzen von HC auf JA-Instanzen von TSP abgebildet werden, und NEIN-Instanzen von HC auf NEIN-Instanzen von TSP.

Sei $G = (V, E)$ eine Instanz von HC.

Wir konstruieren eine Instanz $f(G) = (\tilde{G}, k)$ von TSP wie folgt:

Knoten $\tilde{V} := V$,

Beispiel Polynomialzeitreduktion

Satz: Entscheidungsproblem Hamiltonscher Kreis (HC) $\leq_p$ Entscheidungsproblem Handlungsreisendenproblem (TSP)

Beweis:

Was müssen wir tun, um das zu zeigen?

Wir suchen eine Polynomialzeitreduktion f, die jede Instanz I von HC in eine Instanz $f(I)$ von TSP überführt,
so dass JA-Instanzen von HC auf JA-Instanzen von TSP abgebildet werden, und NEIN-Instanzen von HC auf NEIN-Instanzen von TSP.

Sei $G = (V, E)$ eine Instanz von HC.

Wir konstruieren eine Instanz $f(G) = (\tilde{G}, k)$ von TSP wie folgt:

Knoten $\tilde{V} := V$, Kanten $\tilde{E} := \{\{u, v\} \in V \times V, u \neq v\}$

Beispiel Polynomialzeitreduktion

Satz: Entscheidungsproblem Hamiltonscher Kreis (HC) $\leq_p$ Entscheidungsproblem Handlungsreisendenproblem (TSP)

Beweis:

Was müssen wir tun, um das zu zeigen?

Wir suchen eine Polynomialzeitreduktion f, die jede Instanz I von HC in eine Instanz $f(I)$ von TSP überführt,
so dass JA-Instanzen von HC auf JA-Instanzen von TSP abgebildet werden, und NEIN-Instanzen von HC auf NEIN-Instanzen von TSP.

Sei $G = (V, E)$ eine Instanz von HC.

Wir konstruieren eine Instanz $f(G) = (\tilde{G}, k)$ von TSP wie folgt:

Knoten $\tilde{V} := V$, Kanten $\tilde{E} := \{\{u, v\} \in V \times V, u \neq v\}$ $\tilde{G} := (\tilde{V}, \tilde{E})$

Beispiel Polynomialzeitreduktion

Satz: Entscheidungsproblem Hamiltonscher Kreis (HC) $\leq_p$ Entscheidungsproblem Handlungsreisendenproblem (TSP)

Beweis:

Was müssen wir tun, um das zu zeigen?

Wir suchen eine Polynomialzeitreduktion f, die jede Instanz I von HC in eine Instanz $f(I)$ von TSP überführt,
so dass JA-Instanzen von HC auf JA-Instanzen von TSP abgebildet werden, und NEIN-Instanzen von HC auf NEIN-Instanzen von TSP.

Sei $G = (V, E)$ eine Instanz von HC.

Wir konstruieren eine Instanz $f(G) = (\tilde{G}, k)$ von TSP wie folgt:

Knoten $\tilde{V} := V$, Kanten $\tilde{E} := \{\{u, v\} \in V \times V, u \neq v\}$ $\tilde{G} := (\tilde{V}, \tilde{E})$

Kantenlabel: $l_e = 1$ falls $e \in E$, $l_e = |V| + 1$ falls $e \in \tilde{E} \setminus E$

Beispiel Polynomialzeitreduktion

Satz: Entscheidungsproblem Hamiltonscher Kreis (HC) $\leq_p$ Entscheidungsproblem Handlungsreisendenproblem (TSP)

Beweis:

Was müssen wir tun, um das zu zeigen?

Wir suchen eine Polynomialzeitreduktion f, die jede Instanz I von HC in eine Instanz $f(I)$ von TSP überführt,
so dass JA-Instanzen von HC auf JA-Instanzen von TSP abgebildet werden, und NEIN-Instanzen von HC auf NEIN-Instanzen von TSP.

Sei $G = (V, E)$ eine Instanz von HC.

Wir konstruieren eine Instanz $f(G) = (\tilde{G}, k)$ von TSP wie folgt:

Knoten $\tilde{V} := V$, Kanten $\tilde{E} := \{\{u, v\} \in V \times V, u \neq v\}$ $\quad \tilde{G} := (\tilde{V}, \tilde{E})$

Kantenlabel: $l_e = 1$ falls $e \in E$, $l_e = |V| + 1$ falls $e \in \tilde{E} \setminus E$

$k := |V|$

Beispiel Polynomialzeitreduktion

Satz: Entscheidungsproblem Hamiltonscher Kreis (HC) $\leq_p$ Entscheidungsproblem Handlungsreisendenproblem (TSP)

Beweis:

Was müssen wir tun, um das zu zeigen?

Wir suchen eine Polynomialzeitreduktion f, die jede Instanz I von HC in eine Instanz $f(I)$ von TSP überführt,
so dass JA-Instanzen von HC auf JA-Instanzen von TSP abgebildet werden, und NEIN-Instanzen von HC auf NEIN-Instanzen von TSP.

Sei $G = (V, E)$ eine Instanz von HC.

Wir konstruieren eine Instanz $f(G) = (\tilde{G}, k)$ von TSP wie folgt:

Knoten $\tilde{V} := V$, Kanten $\tilde{E} := \{\{u, v\} \in V \times V, u \neq v\}$ $\tilde{G} := (\tilde{V}, \tilde{E})$

Kantenlabel: $l_e = 1$ falls $e \in E$, $l_e = |V| + 1$ falls $e \in \tilde{E} \setminus E$

$k := |V|$

Ist das eine Reduktion *in Polynomialzeit* ?

Beispiel Polynomialzeitreduktion

Satz: Entscheidungsproblem Hamiltonscher Kreis (HC) $\leq_p$ Entscheidungsproblem Handlungsreisendenproblem (TSP)

Beweis:

Was müssen wir tun, um das zu zeigen?

Wir suchen eine Polynomialzeitreduktion f, die jede Instanz I von HC in eine Instanz $f(I)$ von TSP überführt,
so dass JA-Instanzen von HC auf JA-Instanzen von TSP abgebildet werden, und NEIN-Instanzen von HC auf NEIN-Instanzen von TSP.

Sei $G = (V, E)$ eine Instanz von HC.
Wir konstruieren eine Instanz $f(G) = (\tilde{G}, k)$ von TSP wie folgt:

Knoten $\tilde{V} := V$, Kanten $\tilde{E} := \{\{u, v\} \in V \times V, u \neq v\}$ $\tilde{G} := (\tilde{V}, \tilde{E})$

Kantenlabel: $l_e = 1$ falls $e \in E$, $l_e = |V| + 1$ falls $e \in \tilde{E} \setminus E$

$k := |V|$

Ist G eine JA-Instanz von HC genau dann, wenn $(\tilde{G}, k)$ eine JA-Instanz von TSP ist?

Beispiel Polynomialzeitreduktion

Satz: Entscheidungsproblem Hamiltonscher Kreis (HC) $\leq_p$ Entscheidungsproblem Handlungsreisendenproblem (TSP)

Beweis:

Sei $G = (V, E)$ eine Instanz von HC.

Wir konstruieren eine Instanz $f(G) = (\tilde{G}, k)$ von TSP wie folgt:

Knoten $\tilde{V} := V$, Kanten $\tilde{E} := \{\{u, v\} \in V \times V, u \neq v\}$ $\quad \tilde{G} := (\tilde{V}, \tilde{E})$

Kantenlabel: $l_e = 1$ falls $e \in E$, $l_e = |V| + 1$ falls $e \in \tilde{E} \setminus E$

$k := |V|$

Ist G eine JA-Instanz von HC genau dann, wenn $(\tilde{G}, k)$ eine JA-Instanz von TSP ist?

Beispiel Polynomialzeitreduktion

Satz: Entscheidungsproblem Hamiltonscher Kreis (HC) $\leq_p$ Entscheidungsproblem Handlungsreisendenproblem (TSP)

Beweis:

Sei $G = (V, E)$ eine Instanz von HC.

Wir konstruieren eine Instanz $f(G) = (\tilde{G}, k)$ von TSP wie folgt:

Knoten $\tilde{V} := V$, Kanten $\tilde{E} := \{\{u, v\} \in V \times V, u \neq v\}$ $\quad \tilde{G} := (\tilde{V}, \tilde{E})$

Kantenlabel: $l_e = 1$ falls $e \in E$, $l_e = |V| + 1$ falls $e \in \tilde{E} \setminus E$

$k := |V|$

Ist G eine JA-Instanz von HC genau dann, wenn $(\tilde{G}, k)$ eine JA-Instanz von TSP ist?

Falls G eine JA-Instanz von HC ist,

Beispiel Polynomialzeitreduktion

Satz: Entscheidungsproblem Hamiltonscher Kreis (HC) $\leq_p$ Entscheidungsproblem Handlungsreisendenproblem (TSP)

Beweis:

Sei $G = (V, E)$ eine Instanz von HC.

Wir konstruieren eine Instanz $f(G) = (\tilde{G}, k)$ von TSP wie folgt:

Knoten $\tilde{V} := V$, Kanten $\tilde{E} := \{\{u, v\} \in V \times V, u \neq v\}$ $\quad \tilde{G} := (\tilde{V}, \tilde{E})$

Kantenlabel: $l_e = 1$ falls $e \in E$, $l_e = |V| + 1$ falls $e \in \tilde{E} \setminus E$

$k := |V|$

Ist G eine JA-Instanz von HC genau dann, wenn $(\tilde{G}, k)$ eine JA-Instanz von TSP ist?

Falls G eine JA-Instanz von HC ist,

dann gibt es einen Hamiltonschen Kreis $(v_{(1)}, v_{(2)}, v_{(3)}, \ldots, v_{(n)}, v_{(1)})$ in G

Beispiel Polynomialzeitreduktion

Satz: Entscheidungsproblem Hamiltonscher Kreis (HC) $\leq_p$ Entscheidungsproblem Handlungsreisendenproblem (TSP)

Beweis:

Sei $G = (V, E)$ eine Instanz von HC.

Wir konstruieren eine Instanz $f(G) = (\tilde{G}, k)$ von TSP wie folgt:

Knoten $\tilde{V} := V$, Kanten $\tilde{E} := \{\{u, v\} \in V \times V, u \neq v\}$ $\quad \tilde{G} := (\tilde{V}, \tilde{E})$

Kantenlabel: $l_e = 1$ falls $e \in E$, $l_e = |V| + 1$ falls $e \in \tilde{E} \setminus E$

$k := |V|$

Ist G eine JA-Instanz von HC genau dann, wenn $(\tilde{G}, k)$ eine JA-Instanz von TSP ist?

Falls G eine JA-Instanz von HC ist,

dann gibt es einen Hamiltonschen Kreis $(v_{(1)}, v_{(2)}, v_{(3)}, \ldots, v_{(n)}, v_{(1)})$ in G

$\Rightarrow$ die Kanten $\{v_{(i)}, v_{(i+1)}\}$ für $i = 1, \ldots, n-1$ und $\{v_{(n)}, v_{(1)}\}$ sind in E

Beispiel Polynomialzeitreduktion

Satz: Entscheidungsproblem Hamiltonscher Kreis (HC) $\leq_p$ Entscheidungsproblem Handlungsreisendenproblem (TSP)

Beweis:

Sei $G = (V, E)$ eine Instanz von HC.

Wir konstruieren eine Instanz $f(G) = (\tilde{G}, k)$ von TSP wie folgt:

Knoten $\tilde{V} := V$, Kanten $\tilde{E} := \{\{u, v\} \in V \times V, u \neq v\}$ $\quad \tilde{G} := (\tilde{V}, \tilde{E})$

Kantenlabel: $l_e = 1$ falls $e \in E$, $l_e = |V| + 1$ falls $e \in \tilde{E} \setminus E$

$k := |V|$

Ist G eine JA-Instanz von HC genau dann, wenn $(\tilde{G}, k)$ eine JA-Instanz von TSP ist?

Falls G eine JA-Instanz von HC ist,

dann gibt es einen Hamiltonschen Kreis $(v_{(1)}, v_{(2)}, v_{(3)}, \ldots, v_{(n)}, v_{(1)}))$ in G

$\Rightarrow$ die Kanten $\{v_{(i)}, v_{(i+1)}\}$ für $i = 1, \ldots, n-1$ und $\{v_{(n)}, v_{(1)}\}$ sind in E

$\Rightarrow (v_{(1)}, v_{(2)}, , \ldots, v_{(n)}, v_{(1)}))$ ist Tour (Kreis) mit Länge $1 + \ldots + 1 = |V|$ in $\tilde{G}$

Beispiel Polynomialzeitreduktion

Satz: Entscheidungsproblem Hamiltonscher Kreis (HC) $\leq_p$ Entscheidungsproblem Handlungsreisendenproblem (TSP)

Beweis:

Sei $G = (V, E)$ eine Instanz von HC.

Wir konstruieren eine Instanz $f(G) = (\tilde{G}, k)$ von TSP wie folgt:

Knoten $\tilde{V} := V$, Kanten $\tilde{E} := \{\{u, v\} \in V \times V, u \neq v\}$ $\quad \tilde{G} := (\tilde{V}, \tilde{E})$

Kantenlabel: $l_e = 1$ falls $e \in E$, $l_e = |V| + 1$ falls $e \in \tilde{E} \setminus E$

$k := |V|$

Ist G eine JA-Instanz von HC genau dann, wenn $(\tilde{G}, k)$ eine JA-Instanz von TSP ist?

Falls G eine JA-Instanz von HC ist,
dann gibt es einen Hamiltonschen Kreis $(v_{(1)}, v_{(2)}, v_{(3)}, \ldots, v_{(n)}, v_{(1)}))$ in G
$\Rightarrow$ die Kanten $\{v_{(i)}, v_{(i+1)}\}$ für $i = 1, \ldots, n-1$ und $\{v_{(n)}, v_{(1)}\}$ sind in E

$\Rightarrow$ $(v_{(1)}, v_{(2)}, , \ldots, v_{(n)}, v_{(1)}))$ ist Tour (Kreis) mit Länge $1 + \ldots + 1 = |V|$ in $\tilde{G}$

$\Rightarrow$ $(\tilde{G}, |V|)$ ist eine JA-Instanz von TSP.

Beispiel Polynomialzeitreduktion

Satz: Entscheidungsproblem Hamiltonscher Kreis (HC) $\leq_p$ Entscheidungsproblem Handlungsreisendenproblem (TSP)

Beweis:

Sei $G = (V, E)$ eine Instanz von HC.

Wir konstruieren eine Instanz $f(G) = (\tilde{G}, k)$ von TSP wie folgt:

Knoten $\tilde{V} := V$, Kanten $\tilde{E} := \{\{u, v\} \in V \times V, u \neq v\}$ $\quad \tilde{G} := (\tilde{V}, \tilde{E})$

Kantenlabel: $l_e = 1$ falls $e \in E$, $l_e = |V| + 1$ falls $e \in \tilde{E} \setminus E$

$k := |V|$

Ist G eine JA-Instanz von HC genau dann, wenn $(\tilde{G}, k)$ eine JA-Instanz von TSP ist?

Falls $f(G) = (\tilde{G}, |V|)$ eine JA-Instanz von HC ist,

Beispiel Polynomialzeitreduktion

Satz: Entscheidungsproblem Hamiltonscher Kreis (HC) $\leq_p$ Entscheidungsproblem Handlungsreisendenproblem (TSP)

Beweis:

Sei $G = (V, E)$ eine Instanz von HC.

Wir konstruieren eine Instanz $f(G) = (\tilde{G}, k)$ von TSP wie folgt:

Knoten $\tilde{V} := V$, Kanten $\tilde{E} := \{\{u, v\} \in V \times V, u \neq v\}$ $\quad \tilde{G} := (\tilde{V}, \tilde{E})$

Kantenlabel: $l_e = 1$ falls $e \in E$, $l_e = |V| + 1$ falls $e \in \tilde{E} \setminus E$

$k := |V|$

Ist G eine JA-Instanz von HC genau dann, wenn $(\tilde{G}, k)$ eine JA-Instanz von TSP ist?

Falls $f(G) = (\tilde{G}, |V|)$ eine JA-Instanz von HC ist,
dann gibt es eine TSP-Tour K mit Länge $\sum_{e \in K} l_e \leq |V|$ in $\tilde{G}$

Beispiel Polynomialzeitreduktion

Satz: Entscheidungsproblem Hamiltonscher Kreis (HC) $\leq_p$ Entscheidungsproblem Handlungsreisendenproblem (TSP)

Beweis:

Sei $G = (V, E)$ eine Instanz von HC.

Wir konstruieren eine Instanz $f(G) = (\tilde{G}, k)$ von TSP wie folgt:

Knoten $\tilde{V} := V$, Kanten $\tilde{E} := \{\{u, v\} \in V \times V, u \neq v\}$ $\quad \tilde{G} := (\tilde{V}, \tilde{E})$

Kantenlabel: $l_e = 1$ falls $e \in E$, $l_e = |V| + 1$ falls $e \in \tilde{E} \setminus E$

$k := |V|$

Ist G eine JA-Instanz von HC genau dann, wenn $(\tilde{G}, k)$ eine JA-Instanz von TSP ist?

Falls $f(G) = (\tilde{G}, |V|)$ eine JA-Instanz von HC ist,
dann gibt es eine TSP-Tour K mit Länge $\sum_{e \in K} l_e \leq |V|$ in $\tilde{G}$
$\Rightarrow K$ enthält nur Kanten aus E (da $l_e > |V| \forall e \notin E$)

Beispiel Polynomialzeitreduktion

Satz: Entscheidungsproblem Hamiltonscher Kreis (HC) $\leq_p$ Entscheidungsproblem Handlungsreisendenproblem (TSP)

Beweis:

Sei $G = (V, E)$ eine Instanz von HC.

Wir konstruieren eine Instanz $f(G) = (\tilde{G}, k)$ von TSP wie folgt:

Knoten $\tilde{V} := V$, Kanten $\tilde{E} := \{\{u, v\} \in V \times V, u \neq v\}$ $\quad \tilde{G} := (\tilde{V}, \tilde{E})$

Kantenlabel: $l_e = 1$ falls $e \in E$, $l_e = |V| + 1$ falls $e \in \tilde{E} \setminus E$

$k := |V|$

Ist G eine JA-Instanz von HC genau dann, wenn $(\tilde{G}, k)$ eine JA-Instanz von TSP ist?

Falls $f(G) = (\tilde{G}, |V|)$ eine JA-Instanz von HC ist,
dann gibt es eine TSP-Tour K mit Länge $\sum_{e \in K} l_e \leq |V|$ in $\tilde{G}$
$\Rightarrow K$ enthält nur Kanten aus E (da $l_e > |V| \forall e \notin E$)
$\Rightarrow K$ enthält genau $|V|$ Kanten (da alle Knoten besucht werden und alle Kanten aus E Länge 1 haben)

Beispiel Polynomialzeitreduktion

Satz: Entscheidungsproblem Hamiltonscher Kreis (HC) $\leq_p$ Entscheidungsproblem Handlungsreisendenproblem (TSP)

Beweis:

Sei $G = (V, E)$ eine Instanz von HC.

Wir konstruieren eine Instanz $f(G) = (\tilde{G}, k)$ von TSP wie folgt:

Knoten $\tilde{V} := V$, Kanten $\tilde{E} := \{\{u, v\} \in V \times V, u \neq v\}$ $\quad \tilde{G} := (\tilde{V}, \tilde{E})$

Kantenlabel: $l_e = 1$ falls $e \in E$, $l_e = |V| + 1$ falls $e \in \tilde{E} \setminus E$

$k := |V|$

Ist G eine JA-Instanz von HC genau dann, wenn $(\tilde{G}, k)$ eine JA-Instanz von TSP ist?

Falls $f(G) = (\tilde{G}, |V|)$ eine JA-Instanz von HC ist,
dann gibt es eine TSP-Tour K mit Länge $\sum_{e \in K} l_e \leq |V|$ in $\tilde{G}$
$\Rightarrow$ K enthält nur Kanten aus E (da $l_e > |V| \forall e \notin E$)
$\Rightarrow$ K enthält genau $|V|$ Kanten (da alle Knoten besucht werden und alle Kanten aus E Länge 1 haben)
$\Rightarrow$ K besucht jeden Knoten *genau* einmal

Beispiel Polynomialzeitreduktion

Satz: Entscheidungsproblem Hamiltonscher Kreis (HC) $\leq_p$ Entscheidungsproblem Handlungsreisendenproblem (TSP)

Beweis:

Sei $G = (V, E)$ eine Instanz von HC.

Wir konstruieren eine Instanz $f(G) = (\tilde{G}, k)$ von TSP wie folgt:

Knoten $\tilde{V} := V$, Kanten $\tilde{E} := \{\{u, v\} \in V \times V, u \neq v\}$ $\quad \tilde{G} := (\tilde{V}, \tilde{E})$

Kantenlabel: $l_e = 1$ falls $e \in E$, $l_e = |V| + 1$ falls $e \in \tilde{E} \setminus E$

$k := |V|$

Ist G eine JA-Instanz von HC genau dann, wenn $(\tilde{G}, k)$ eine JA-Instanz von TSP ist?

Falls $f(G) = (\tilde{G}, |V|)$ eine JA-Instanz von HC ist,
dann gibt es eine TSP-Tour K mit Länge $\sum_{e \in K} l_e \leq |V|$ in $\tilde{G}$
$\Rightarrow$ K enthält nur Kanten aus E (da $l_e > |V| \forall e \notin E$)
$\Rightarrow$ K enthält genau $|V|$ Kanten (da alle Knoten besucht werden und alle Kanten aus E Länge 1 haben)
$\Rightarrow$ K besucht jeden Knoten *genau* einmal
$\Rightarrow$ in G ist K ein Hamiltonscher Kreis

Beispiel Polynomialzeitreduktion

Satz: Entscheidungsproblem Hamiltonscher Kreis (HC) $\leq_p$ Entscheidungsproblem Handlungsreisendenproblem (TSP)

Korollar: Falls HC NP-vollständig ist, ist auch TSP NP-vollständig.

Satz von Cook und Levin

Erfüllbarkeitsproblem (Englisch: Satisfiability, kurz: **SAT**)
Gegeben: Menge logischer Variablen X, Formeln (Disjunktionen) F über X
Gesucht: Zuweisung von Werten $t : X \to \{\text{wahr,falsch}\}$ so dass jede Formel wahr ist

Satz von Cook und Levin
SAT ist NP-vollständig.

Satz von Cook und Levin

Erfüllbarkeitsproblem (Englisch: Satisfiability, kurz: **SAT**)
Gegeben: Menge logischer Variablen X, Formeln (Disjunktionen) F über X
Gesucht: Zuweisung von Werten $t : X \to \{\text{wahr,falsch}\}$ so dass jede Formel wahr ist

Satz von Cook und Levin
SAT ist NP-vollständig.

im Beweis wird gezeigt, dass jedes Programm auf einer nichtdeterministischen Turingmaschine (siehe alternative Definition der Klasse NP) auf SAT reduziert werden kann

NP-vollständig und NP-schwer

Ein Entscheidungsproblem X heißt **NP-vollständig**, wenn

1. es in NP liegt und
2. wenn ein polynomieller Lösungsalgorithmus für dieses Problem uns ermöglichen würde, *jedes* andere Problem in NP in Polynomialzeit zu lösen.

Es reicht zu zeigen: für *ein* NP-vollständiges Problem Y gibt es eine Polynomialzeitreduktion auf X

NP-vollständig und NP-schwer

Ein Entscheidungsproblem X heißt **NP-vollständig**, wenn

1. es in NP liegt und
2. wenn ein polynomieller Lösungsalgorithmus für dieses Problem uns ermöglichen würde, *jedes* andere Problem in NP in Polynomialzeit zu lösen.

Es reicht zu zeigen: für *ein* NP-vollständiges Problem Y gibt es eine Polynomialzeitreduktion auf X

zum Beispiel SAT

NP-vollständig und NP-schwer

Ein Entscheidungsproblem X heißt **NP-vollständig**, wenn

1. es in NP liegt und
2. wenn ein polynomieller Lösungsalgorithmus für dieses Problem uns ermöglichen würde, *jedes* andere Problem in NP in Polynomialzeit zu lösen.

Es reicht zu zeigen: für *ein* NP-vollständiges Problem Y gibt es eine Polynomialzeitreduktion auf X

zum Beispiel SAT
oder ein anderes Entscheidungsproblem, für das schonmal jemand gezeigt hat, dass es NP-vollständig ist

NP-vollständig und NP-schwer

Ein Entscheidungsproblem X heißt **NP-vollständig**, wenn

1. es in NP liegt und
2. wenn ein polynomieller Lösungsalgorithmus für dieses Problem uns ermöglichen würde, *jedes* andere Problem in NP in Polynomialzeit zu lösen.

Ein (Entscheidungs- oder Optimierungs-)Problem heißt **NP-schwer**, wenn

- wenn ein polynomieller Lösungsalgorithmus für dieses Problem uns ermöglichen würde, *jedes* Problem in NP in Polynomialzeit zu lösen.

NP-vollständig und NP-schwer

Ein Entscheidungsproblem X heißt **NP-vollständig**, wenn

1. es in NP liegt und
2. wenn ein polynomieller Lösungsalgorithmus für dieses Problem uns ermöglichen würde, *jedes* andere Problem in NP in Polynomialzeit zu lösen.

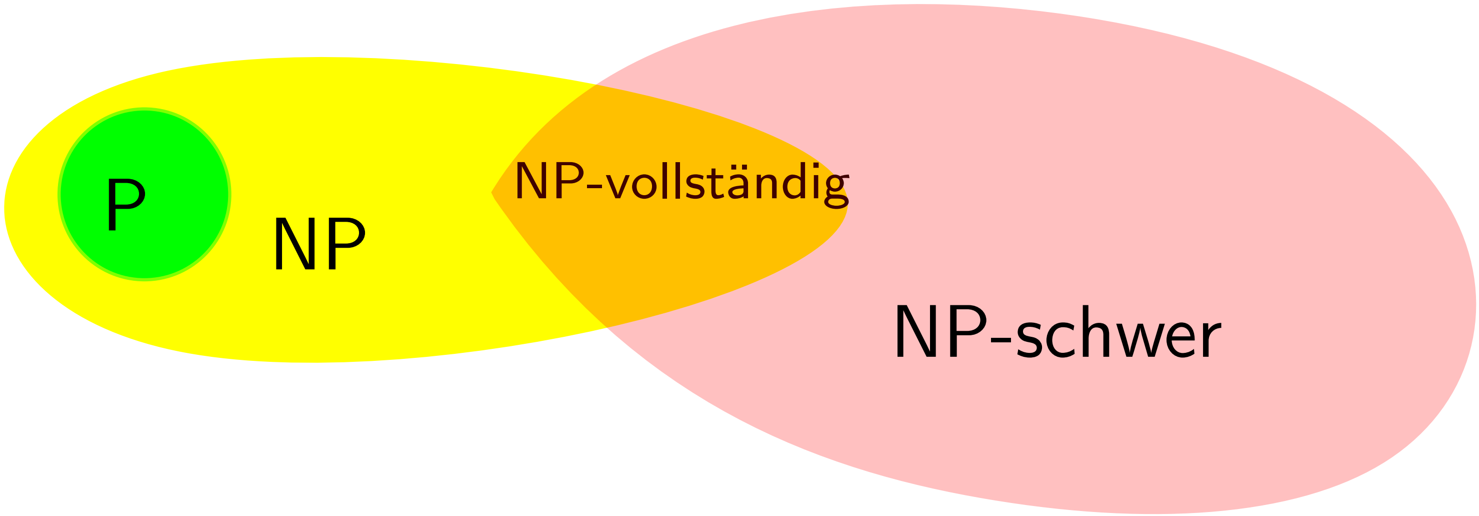
Ein (Entscheidungs- oder Optimierungs-)Problem heißt **NP-schwer**, wenn

- wenn ein polynomieller Lösungsalgorithmus für dieses Problem uns ermöglichen würde, *jedes* Problem in NP in Polynomialzeit zu lösen.

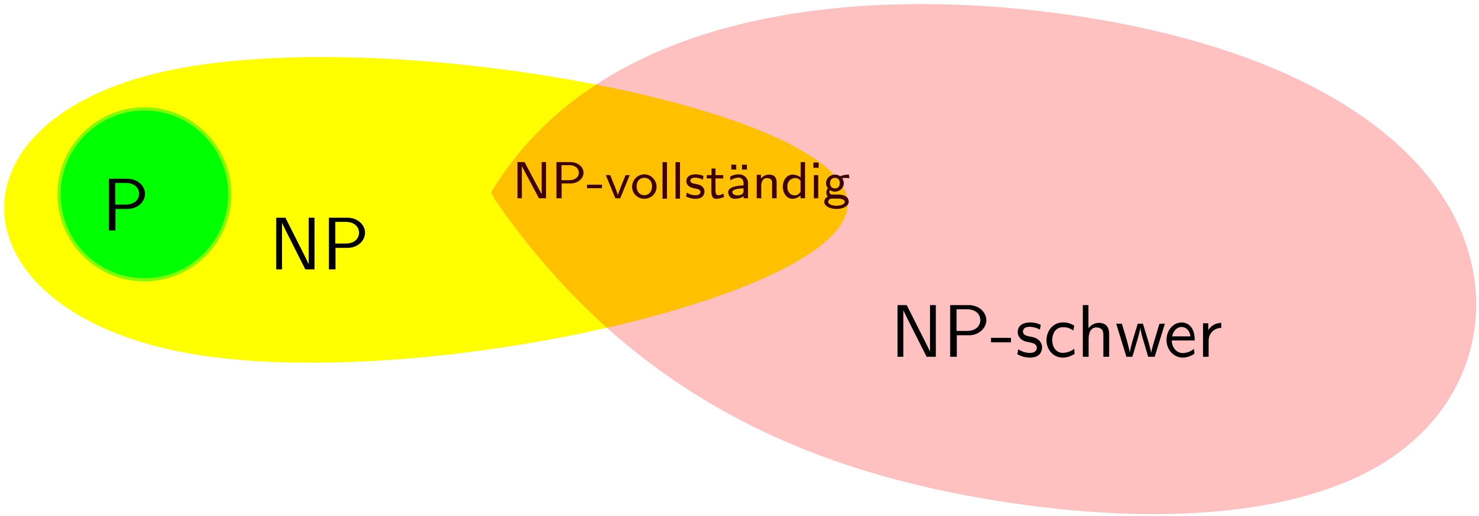
Es reicht zu zeigen: für *ein* NP-schweres Problem Y gibt es eine Polynomialzeitreduktion auf X

zum Beispiel SAT
oder ein anderes Entscheidungsproblem, für das schonmal jemand gezeigt hat, dass es NP-schwer ist

NP-vollständig und NP-schwer

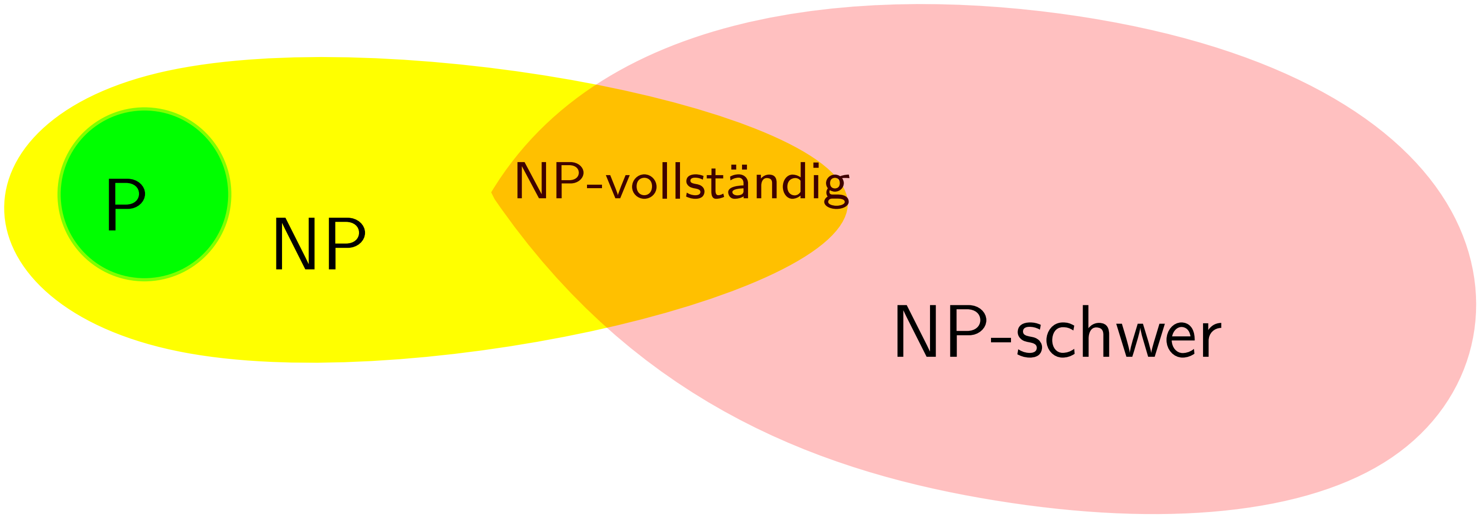


NP-vollständig und NP-schwer



Falls Sie einen polynomiellen Algorithmus für ein NP-schweres Problem finden, dann haben Sie bewiesen, dass $P = NP$!

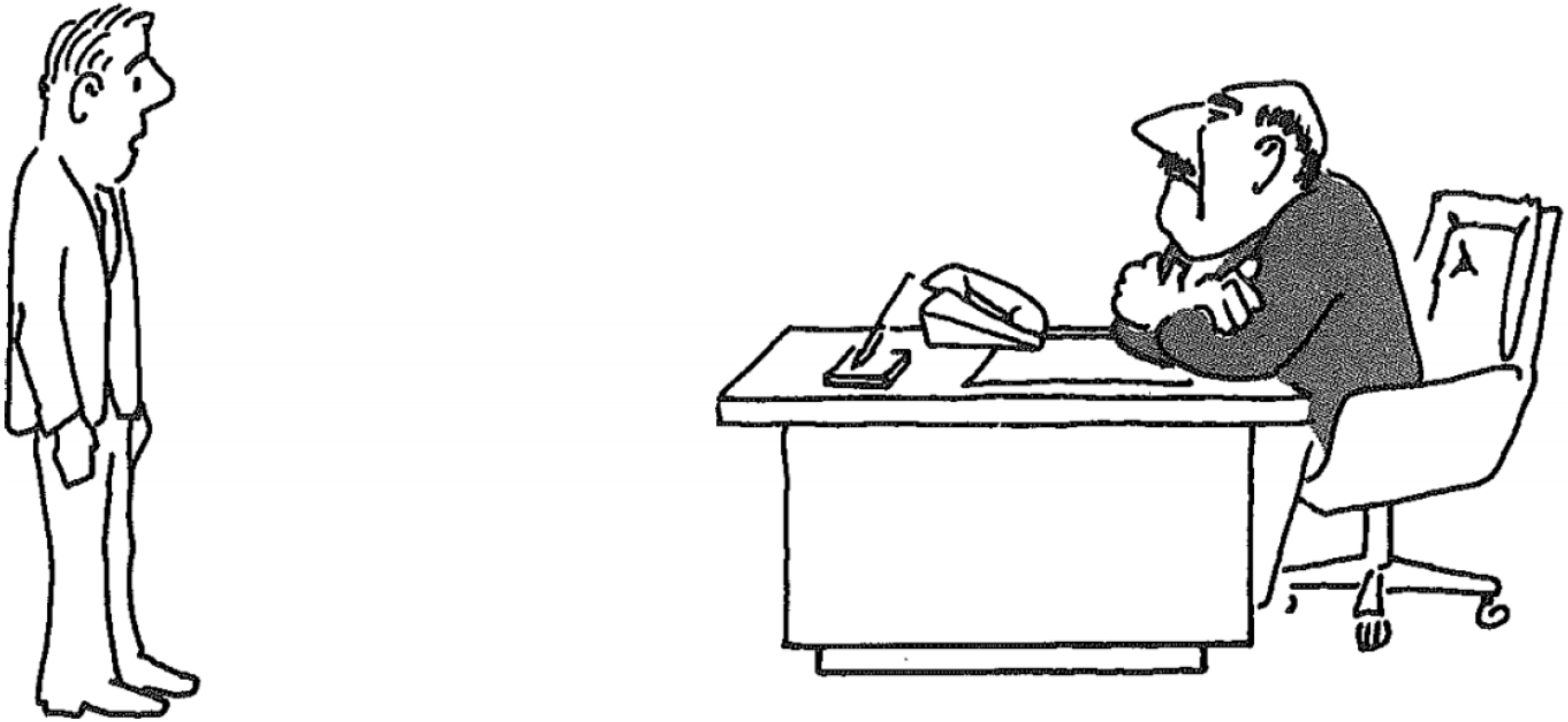
NP-vollständig und NP-schwer



Falls Sie einen polynomiellen Algorithmus für ein NP-schweres Problem finden, dann haben Sie bewiesen, dass $P = NP$!

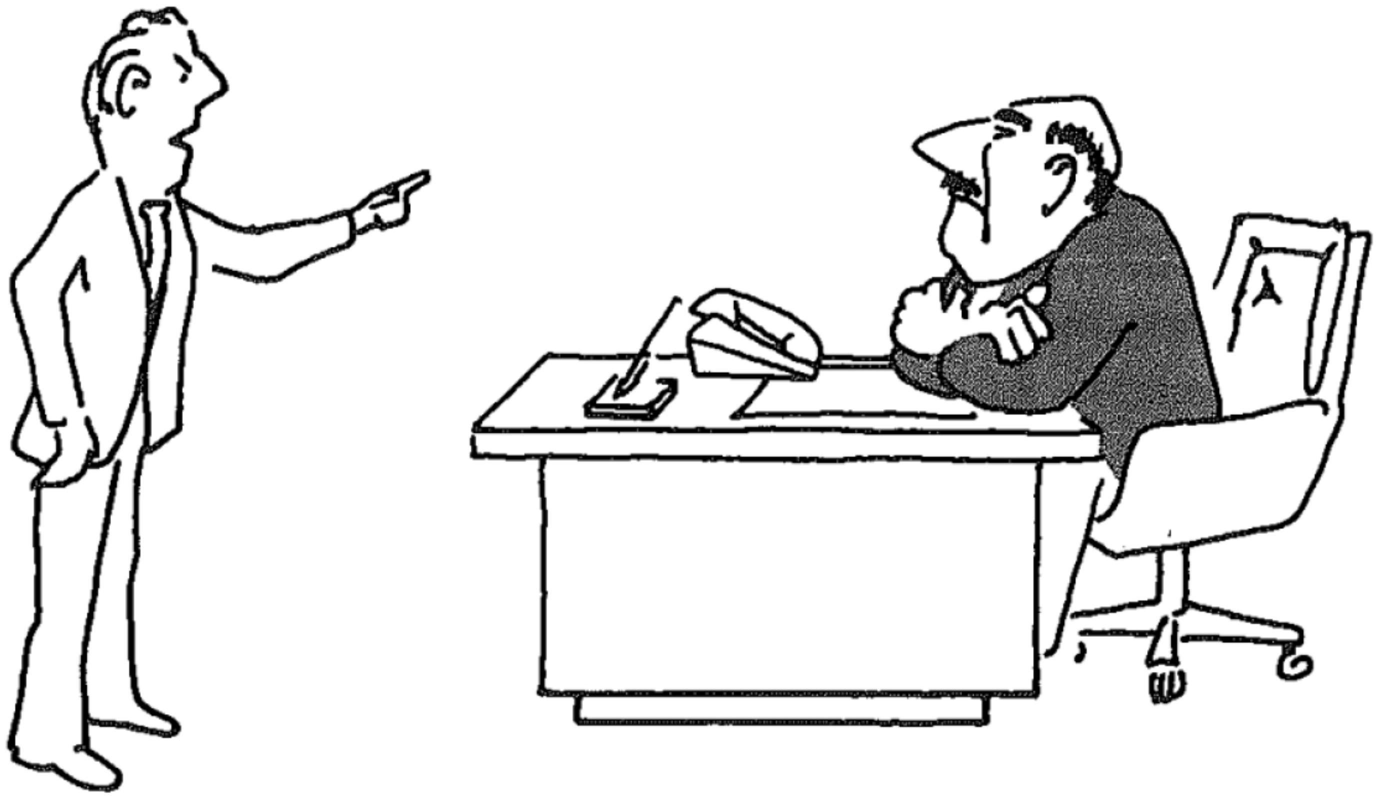
...oder Sie haben sich vertan

Unangenehme Situation...



"I can't find an efficient algorithm, I guess I'm just too dumb."

. . . so wäre es angenehmer



"I can't find an efficient algorithm, because no such algorithm is possible!"

Können wir beweisen, dass es für ein Problem keinen effizienten exakten Algorithmus gibt?

Das macht ein NP-Schwere-Beweis

"I can't find an efficient algorithm, but neither can all these famous people."

Entscheidungsproblem 'Clique'
Gegeben: Ein Graph G, eine Zahl k
Frage: Gibt es in Graph G eine Clique von Größe k

Entscheidungsproblem 'Clique'
Gegeben: Ein Graph G, eine Zahl k
Frage: Gibt es in Graph G eine Clique von Größe k

Satz: Entscheidungsproblem Clique ist NP-vollständig.

Entscheidungsproblem 'Clique'
Gegeben: Ein Graph G, eine Zahl k
Frage: Gibt es in Graph G eine Clique von Größe k

Satz: Entscheidungsproblem Clique ist NP-vollständig.

Beweis: 1. CLIQUE ist in NP.

Was ist das in Polynomialzeit überprüfbare Zertifikat?

Entscheidungsproblem 'Clique'
Gegeben: Ein Graph G, eine Zahl k
Frage: Gibt es in Graph G eine Clique von Größe k

Satz: Entscheidungsproblem Clique ist NP-vollständig.

Beweis: 1. CLIQUE ist in NP.

Was ist das in Polynomialzeit überprüfbare Zertifikat?

2. Eine Reduktion von SAT auf Clique finden Sie zum Beispiel hier: https://tomvanderzanden.nl/LNMB-AC/ac_lecture2_2021.pdf (Slide 55-70).

'Unabhängige Menge' ist NP-vollständig

Entscheidungsproblem 'Unabhängige Menge'
Gegeben: Ein Graph G, eine Zahl k
Frage: Gibt es in Graph G eine unabhängige Menge von Größe k?

Satz: Entscheidungsproblem 'Unabhängige Menge' ist NP-vollständig.

'Unabhängige Menge' ist NP-vollständig

Entscheidungsproblem 'Unabhängige Menge'
Gegeben: Ein Graph G, eine Zahl k
Frage: Gibt es in Graph G eine unabhängige Menge von Größe k?

Satz: Entscheidungsproblem 'Unabhängige Menge' ist NP-vollständig.

Beweis: 'Unabbhängige' Menge ist in NP. ✓

'Unabhängige Menge' ist NP-vollständig

Entscheidungsproblem 'Unabhängige Menge'
Gegeben: Ein Graph G, eine Zahl k
Frage: Gibt es in Graph G eine unabhängige Menge von Größe k?

Satz: Entscheidungsproblem 'Unabhängige Menge' ist NP-vollständig.

Beweis: 'Unabbhängige' Menge ist in NP. ✓

Zeige: Entscheidungsproblem 'Clique' $\leq_p$ Entscheidungsproblem 'Unabhängige Menge'

'Unabhängige Menge' ist NP-vollständig

Entscheidungsproblem 'Unabhängige Menge'
Gegeben: Ein Graph G, eine Zahl k
Frage: Gibt es in Graph G eine unabhängige Menge von Größe k?

Satz: Entscheidungsproblem 'Unabhängige Menge' ist NP-vollständig.

Beweis: 'Unabbhängige' Menge ist in NP. ✓

Zeige: Entscheidungsproblem 'Clique' $\leq_p$ Entscheidungsproblem 'Unabhängige Menge'
Sei (G, k) eine Instanz von Clique.

'Unabhängige Menge' ist NP-vollständig

Entscheidungsproblem 'Unabhängige Menge'
Gegeben: Ein Graph G, eine Zahl k
Frage: Gibt es in Graph G eine unabhängige Menge von Größe k?

Satz: Entscheidungsproblem 'Unabhängige Menge' ist NP-vollständig.

Beweis: 'Unabbhängige' Menge ist in NP. ✓

Zeige: Entscheidungsproblem 'Clique' $\leq_p$ Entscheidungsproblem 'Unabhängige Menge'
Sei (G, k) eine Instanz von Clique.

Konstruiere Instanz von 'Unabhängige Menge' als $(\bar{G}, k)$ ($\bar{G}$ ist der Komplementgraph)

'Unabhängige Menge' ist NP-vollständig

Entscheidungsproblem 'Unabhängige Menge'
Gegeben: Ein Graph G, eine Zahl k
Frage: Gibt es in Graph G eine unabhängige Menge von Größe k?

Satz: Entscheidungsproblem 'Unabhängige Menge' ist NP-vollständig.

Beweis: 'Unabbhängige' Menge ist in NP. ✓

Zeige: Entscheidungsproblem 'Clique' $\leq_p$ Entscheidungsproblem 'Unabhängige Menge'
Sei (G, k) eine Instanz von Clique.

Konstruiere Instanz von 'Unabhängige Menge' als $(\bar{G}, k)$ ($\bar{G}$ ist der Komplementgraph)

Wir haben schon gesehen (VL 5) dass
G hat eine Clique der Größe $k \Leftrightarrow \bar{G}$ hat eine unabhängige Menge der Größe k

'Unabhängige Menge' ist NP-vollständig

Entscheidungsproblem 'Unabhängige Menge'
Gegeben: Ein Graph G, eine Zahl k
Frage: Gibt es in Graph G eine unabhängige Menge von Größe k?

Satz: Entscheidungsproblem 'Unabhängige Menge' ist NP-vollständig.

Beweis: 'Unabbhängige' Menge ist in NP. ✓

Zeige: Entscheidungsproblem 'Clique' $\leq_p$ Entscheidungsproblem 'Unabhängige Menge'
Sei (G, k) eine Instanz von Clique.

Konstruiere Instanz von 'Unabhängige Menge' als $(\bar{G}, k)$ ($\bar{G}$ ist der Komplementgraph)

Wir haben schon gesehen (VL 5) dass
G hat eine Clique der Größe $k \Leftrightarrow \bar{G}$ hat eine unabhängige Menge der Größe k

Daraus folgt: (G, k) ist eine JA-Instanz von 'Clique' $\Leftrightarrow (\bar{G}, k)$ ist eine JA-Instanz von 'Unabhängige Menge' ✓

Noch mehr NP-vollständige Probleme. . .

Satz: Die (Entscheidungsvarianten von) folgenden Probleme sind NP-vollständig: Hamiltonscher Pfad, Hamiltonscher Kreis, Handlungsreisendenproblem, bikriterielle kürzeste Wege, Eckenüberdeckung, Färbungsproblem. . .- und viele mehr

Noch mehr NP-vollständige Probleme. . .

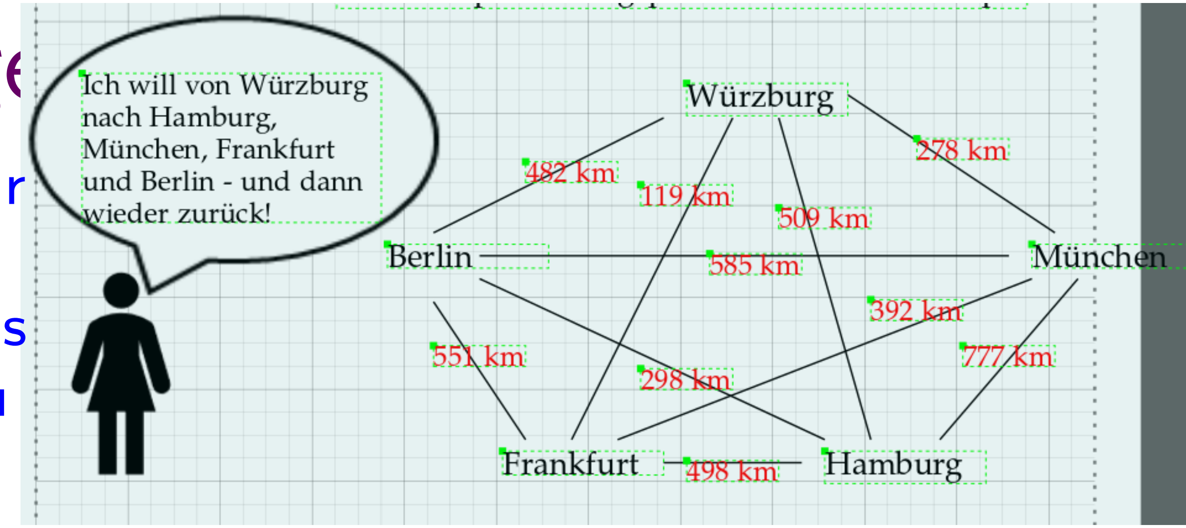
Satz: Die (Entscheidungsvarianten von) folgenden Probleme sind NP-vollständig: Hamiltonscher Pfad, Hamiltonscher Kreis, Handlungsreisendenproblem, bikriterielle kürzeste Wege, Eckenüberdeckung, Färbungsproblem. . . - und viele mehr

Beweis: Zugehörigkeit zu NP sieht man schnell.
Für Reduktionen, die NP-Schwere beweisen siehe zum Beispiel

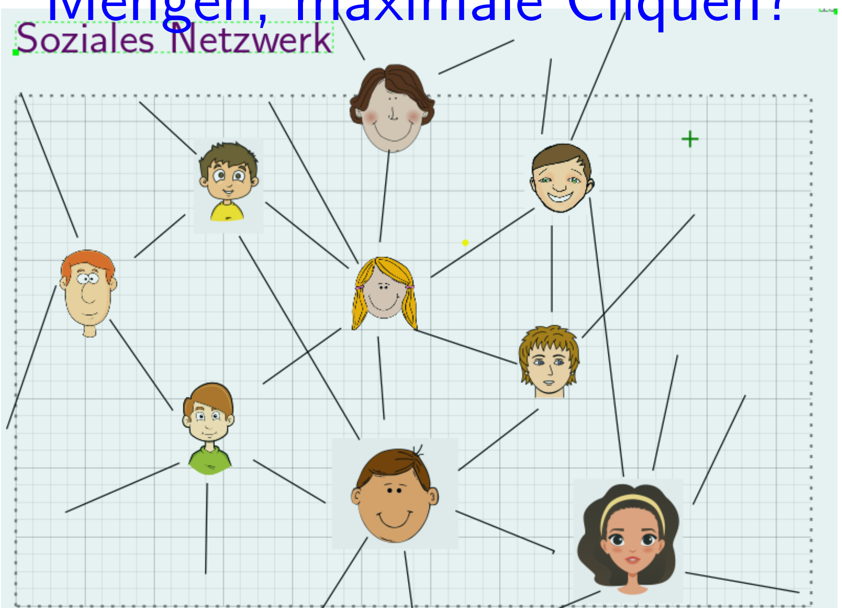
- Reduktionen auf https://tomvanderzanden.nl/LNMB-AC/ (unter Punkt 20-Sep-2021)
- Garey&Johnson: Computers and Intractability (Unibibliothek, leider nicht als Ebook)

(Noch) offene Frage

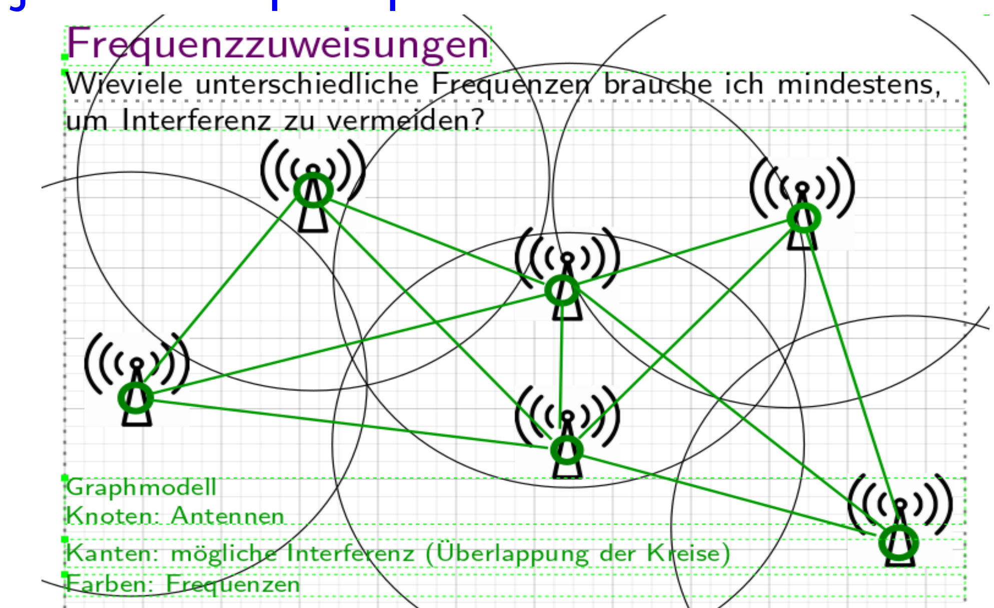
Wie finden wir eine optimale Tour
für Handlungsreisende?
Und was ist ein guter Algorithmus
um einen Hamiltonschen Kreis zu
finden?



Wie finden wir minimale
Eckenüberdeckungen,
maximale unabhängige
Mengen, maximale Cliquen?

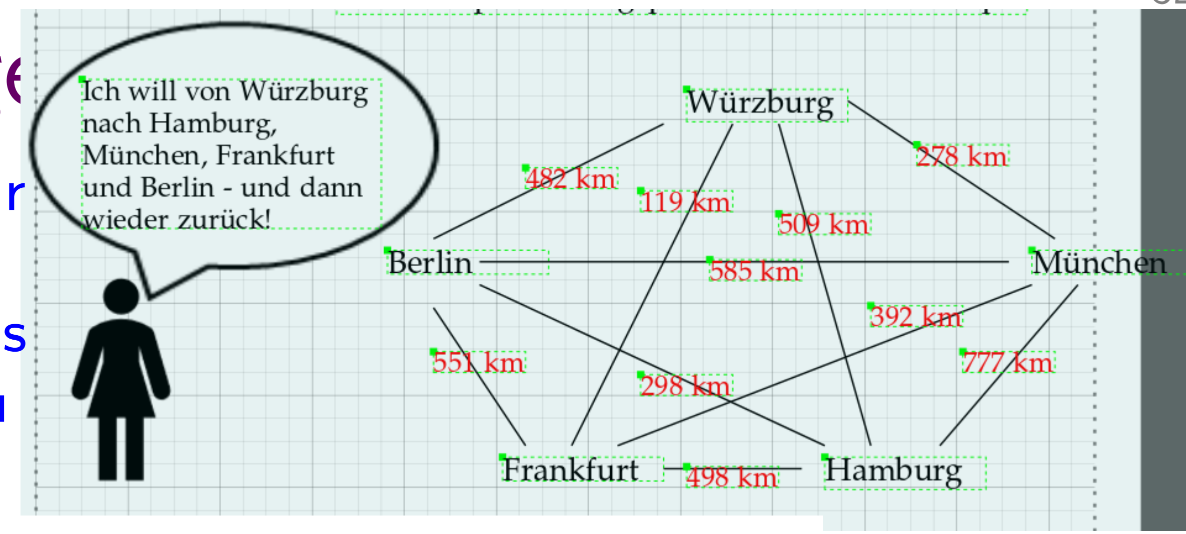


Können wir mit einer Kombination aus
Greedy-Algorithmus und Breitensuche
jeden Graph optimal färben?



(Noch) offene Frage

Wie finden wir eine optimale Tour für Handlungsreisende?
Und was ist ein guter Algorithmus um einen Hamiltonschen Kreis zu finden?

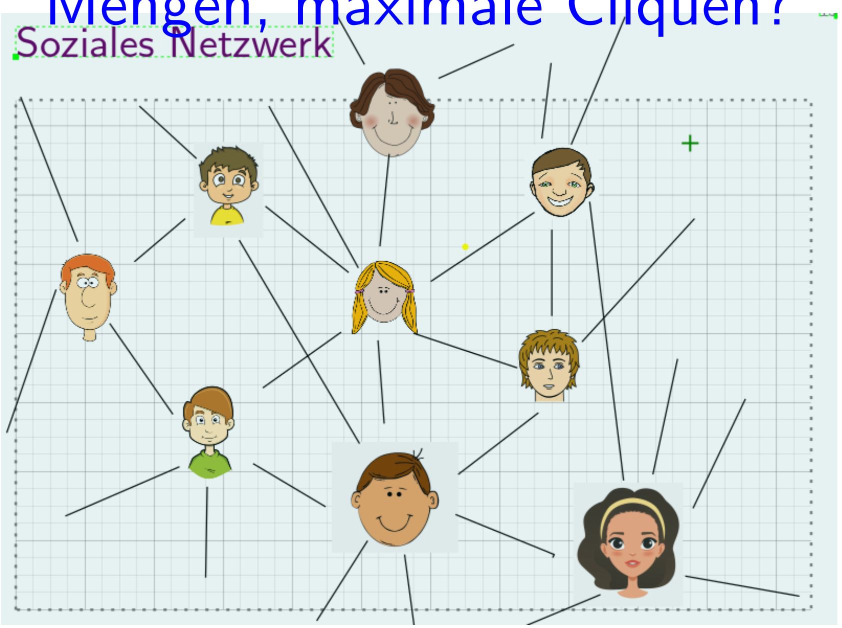


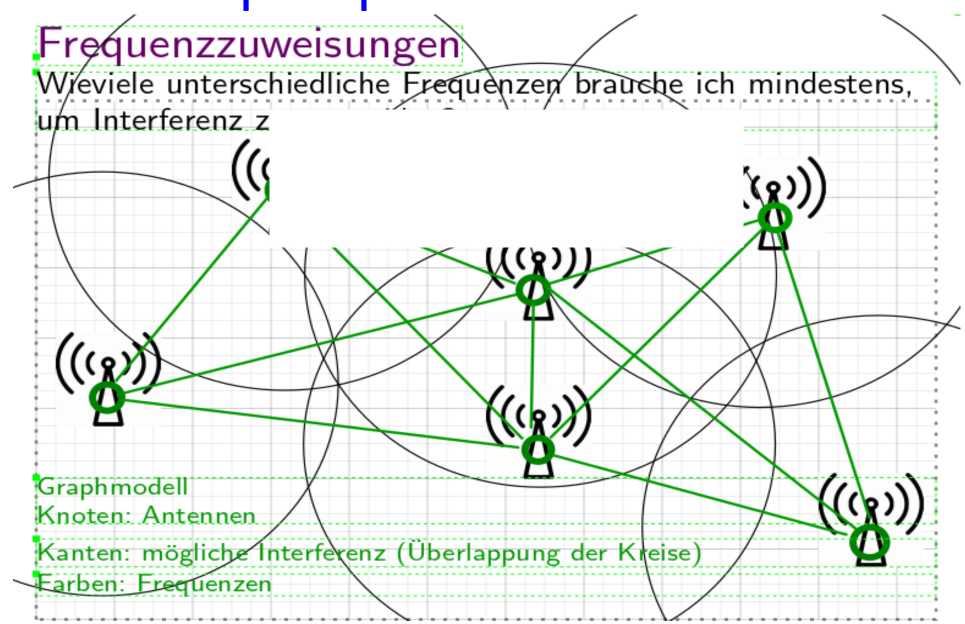
Wie finde… Eckenübe… maximale… …ination aus …itensuche

Falls $P \neq NP$:
Keine polynomiellen Algorithmen möglich.

Mengen, maximale Cliquen?

jeden Graph optimal färben?





Stichworte heute

Algorithmen:Brute Force

Komplexität: Entscheidungsproblem, P, NP, Reduktion, NP-vollständig, NP-schwer

"I can't find an efficient algorithm, but neither can all these famous people."

Stichworte heute

Algorithmen:Brute Force

Komplexität: Entscheidungsproblem, P, NP, Reduktion, NP-vollständig, NP-schwer

Nächste Woche: Was tun wir, wenn der Chef damit nicht zufrieden ist?

"I can't find an efficient algorithm, but neither can all these famous people."

Stichworte heute

Algorithmen:Brute Force

Komplexität: Entscheidungsproblem, P, NP, Reduktion, NP-vollständig, NP-schwer

Nächste Woche: Was tun wir, wenn der Chef damit nicht zufrieden ist?

"I can't find an efficient algorithm, but neither can all these famous people."

Wer noch mehr über Komplexität lernen will, die/der ist in den Vorlesungen von Prof. Glaßer gut aufgehoben (oder schaut einmal in die Videos von Tom van der Zanden und mir auf https://tomvanderzanden.nl/LNMB AC/)