Technische
Universität
Braunschweig

Algorithmen und Datenstrukturen II

8. Vorlesung

Linda Kleist, 19.06.2019

Komplexität von Knapsack und 3-SAT

Problem 9

3-SAT $\in$ P?

Satz 10

Wenn 01-KNAPSACK $\in P$ ist, dann ist auch 3-SAT $\in P$.

Beweis.

Tafel... $\square$

Reduktion

Ein Entscheidungsproblem A lässt sich zu einem Entscheidungsproblem B reduzieren, wenn

- ein Polynomialzeit-Algorithmus existiert, der
- jede Instanz I_A von Problem A in eine Instanz I_B von B transformiert,
- sodass I_A genau dann wahr ist wenn I_B wahr ist.

Wir schreiben $A \leq_p B$.

Beispiele

- $\textsc{Subset Sum} \leq_p$ 01-$\textsc{Knapsack}$
- 3-$\textsc{Sat} \leq_p \textsc{Subset Sum}$

- Für drei Entscheidungsprobleme A, B, C gilt Transitivität:

$$(A \leq_p B) \wedge (B \leq_p C) \implies A \leq_p C$$

- Aus den Beispielen folgt also: 3-$\textsc{Sat} \leq_p$ 01-$\textsc{Knapsack}$

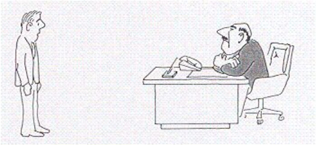
Reduzierbarkeit auf 3-SAT

Satz von Cook und Levin

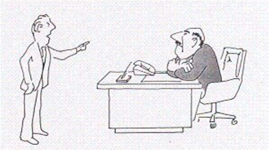
Jedes Problem aus NP lässt sich auf 3-SAT reduzieren, das heißt für jedes Problem A in NP gilt: A $\leq_p$ 3-SAT.

- 3-SAT ist also ein schwerstes Problem in NP.
- Es gilt also auch 01-KNAPSACK $\leq_p$ 3-SAT.
- Wenn 3-SAT in P ist, dann gilt P = NP.
- Erinnerung: Wenn 01-KNAPSACK $\in P$ ist, dann ist auch 3-SAT $\in P$. (Satz 10)

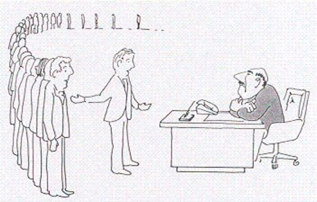
How to explain to your boss as to why your program is so slow...



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



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



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

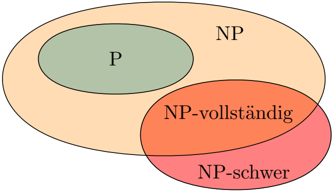
Komplexitätsklassen II

Ein Entscheidungsproblem Prob heißt NP-schwer, wenn sich 3-Sat auf Prob reduzieren lässt, d.h. wenn 3-Sat $\leq_p$ Prob gilt.

Ein Entscheidungsproblem Prob heißt NP-vollständig, wenn es sowohl NP-schwer als auch in NP ist.

- 3-Sat ist NP-vollständig.
- 01-Knapsack ist NP-vollständig.
- SubsetSum ist NP-vollständig.

Falls P $\neq$ NP:



Der Hammer

Falls ein (einziges!) NP-schweres Problem in P ist, gilt P = NP.

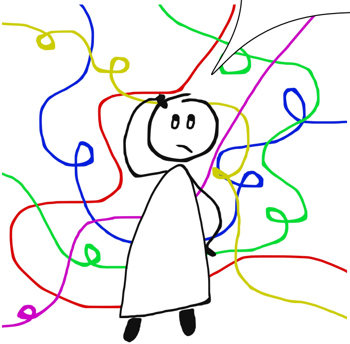
Moment mal...

Maximum Knapsack

dynamisches Programm

pseudopolynomiell

P = NP ?



Komplexitätsklassen III

Ein NP-schweres Entscheidungsproblem PROB heißt schwach NP-schwer, wenn ein Algorithmus existiert, der das Problem in polynomieller Zeit in der Eingabegröße (in ganzen Zahlen) löst.

- 01-KNAPSACK ist schwach NP-schwer.

Ein Entscheidungsproblem PROB heißt stark NP-schwer, wenn es NP-schwer bleibt selbst wenn die Eingabe (alle numerischen Eingabedaten) polynomiell in der Länge des Inputs sind.

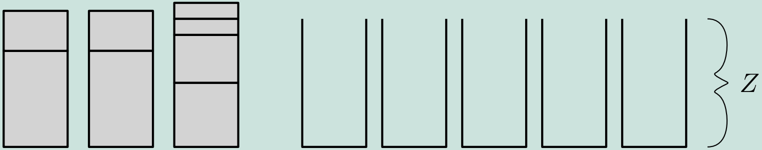
Inapproximierbarkeit

Problem 10 (Bin Packing)

Gegeben:
- Behälter der Größe Z
- Objekte $\{O_1, \ldots, O_n\}$, je mit Größe z_i mit $z_i \leq Z$

Gesucht: Aufteilung der Objekte in minimale Anzahl von Behältern, d.h., eine Zerlegung der Menge $\{1, \ldots, n\}$ in $S_1, S_2, \ldots, S_k$ mit $\sum_{i \in S_j} z_i \leq Z$ so, dass k minimal.

Beispiel



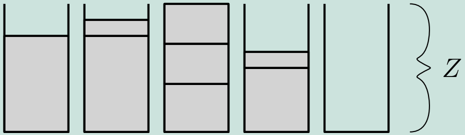
Inapproximierbarkeit

Problem 10 (Bin Packing)

Gegeben:
- Behälter der Größe Z
- Objekte $\{O_1, \ldots, O_n\}$, je mit Größe z_i mit $z_i \leq Z$

Gesucht: Aufteilung der Objekte in minimale Anzahl von Behältern, d.h., eine Zerlegung der Menge $\{1, \ldots, n\}$ in $S_1, S_2, \ldots, S_k$ mit $\sum_{i \in S_j} z_i \leq Z$ so, dass k minimal.

Beispiel



Inapproximierbarkeit

Problem 10 (Bin Packing)

Gegeben: - Behälter der Größe Z
- Objekte $\{O_1, \ldots, O_n\}$, je mit Größe z_i mit $z_i \leq Z$

Gesucht: Aufteilung der Objekte in minimale Anzahl von Behältern, d.h., eine Zerlegung der Menge $\{1, \ldots, n\}$ in $S_1, S_2, \ldots, S_k$ mit $\sum_{i \in S_j} z_i \leq Z$ so, dass k minimal.

Satz 11

Falls ein c-Approximationsalgorithmus für Bin Packing mit $c < \frac{3}{2}$ existiert, dann gilt P $=$ NP.

Beweis.

Tafel... $\square$