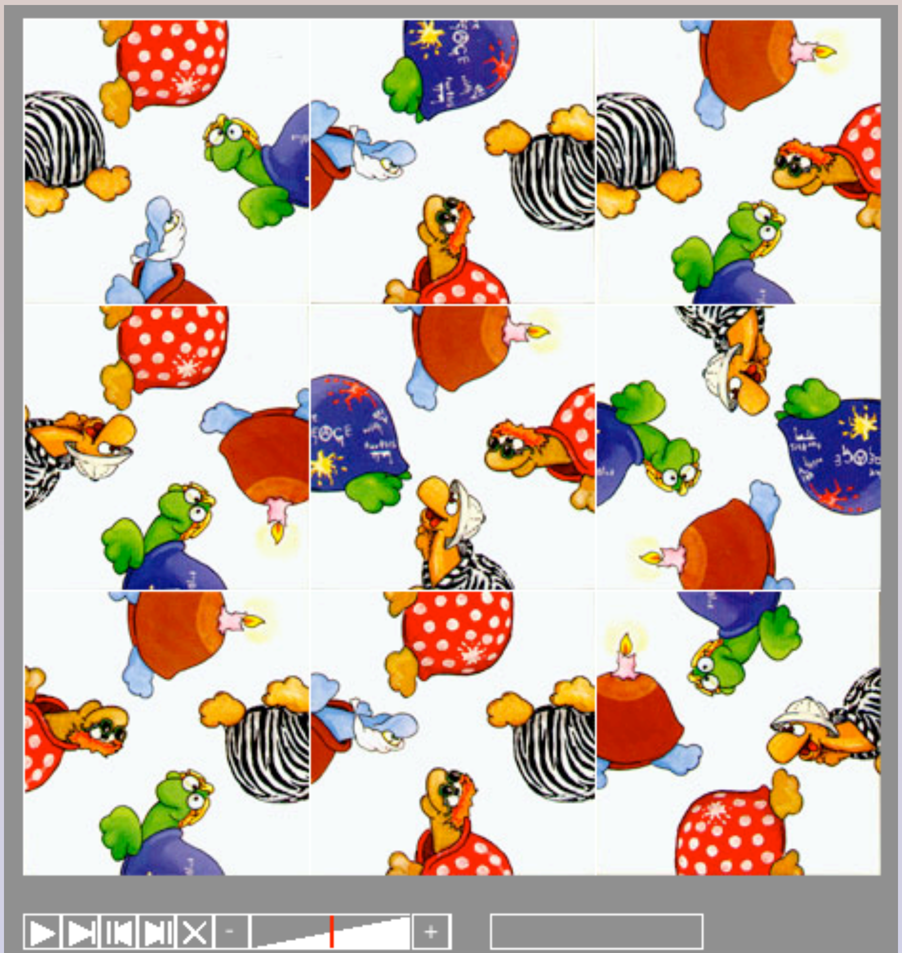
educational engineering lab
Department for Information Technology
University of Zurich



Komplexitätstheorie

Eigenschaften von Algorithmen

- deterministisch - nichtdeterministisch
- sequentiell - parallel
- endlich - unendlich
- reversibel - irreversibel

Ordnung einer Funktion O(f)

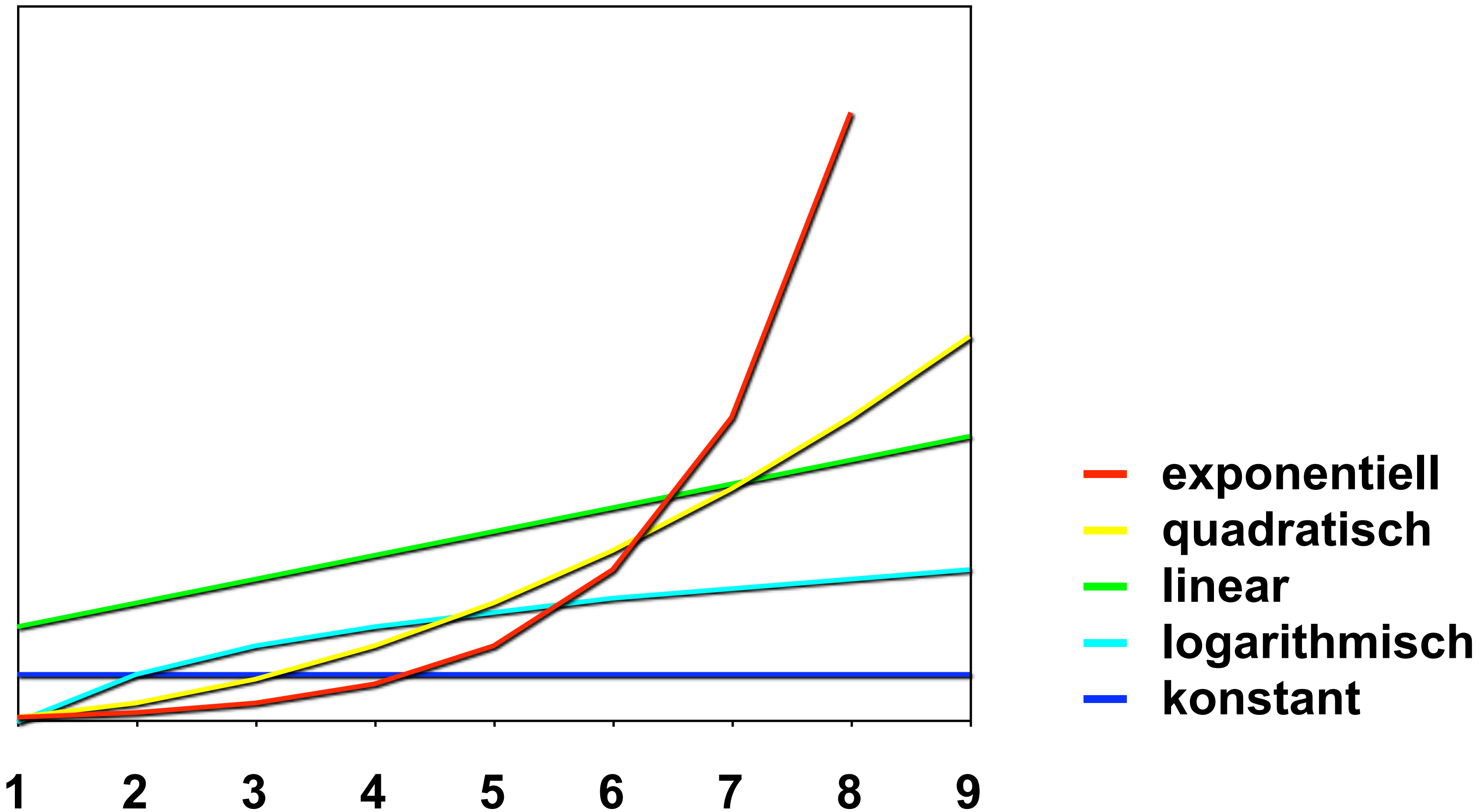
$$f \in O(g) \Leftrightarrow \text{Ex } c,n_0\text{: } c>0 \text{ :(All n: } n \geq n_0 \text{ : } f(n) \leq c*g(n))$$

oder

$$f \in O(g) \Leftrightarrow \lim_{n \to \infty} f(n)/g(n) = c$$

$f \in O(g)$ ... „f ist von der Ordnung g“

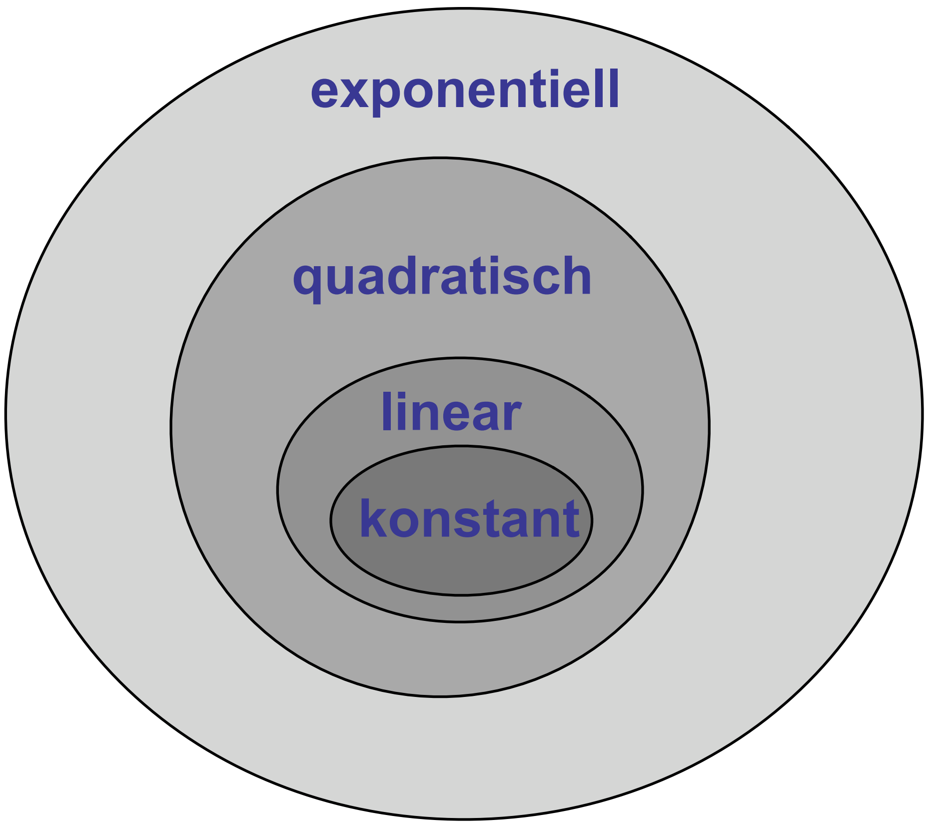
Typische Ordnungen von Funktionen



Bezeichnungen

$O(1)$	konstant
$O(\log n)$	logarithmisch
$O(n)$	linear
$O(n^2)$	quadratisch
$O(n^k)$	polynomial
$O(e^n)$	exponentiell

Mengendarstellung der Ordnung von Funktionen



n	ld n	n ld n	n^2	2^n	n!
10	3	33	100	1024	$3*10^6$
20	4	86	400	10^6	$2*10^{18}$
100	7	664	10'000	10^{31}	10^{161}
1000	10	10'000	10^6		
10000	13	130'000	10^8		

zum Vergleich: es gibt etwa 10^{79} Protonen im Universum

n	ld n	n ld n	n^2	2^n	n!
10	3 µs	33 µs	100 µs	1 ms	3 s
20	4 µs	86 µs	400 µs	1 s	10^5 Jahre
100	7 µs	664 µs	10 ms	10^{17} Jahre	10^{45} Jahre
1000	10 µs	10 ms	1 s		
10000	13 µs	130 ms	100 s		

zum Vergleich: der Urknall war vor etwa 15 Milliarden Jahren

Regeln

$O(c*f(n)) = O(f(n))$

$O(f(n)+g(n)) = \max(O(f(n)),O(g(n)))$

$O(f(n)) \leq O(g(n)) \Leftrightarrow f(n) \in O(g(n))$

$O(f(n)) = O(g(n)) \Leftrightarrow (O(f(n)) \leq O(g(n)))$ and $(O(g(n)) \leq O(f(n)))$

$O(f(n)) < O(g(n)) \Leftrightarrow (O(f(n)) \leq O(g(n)))$ and $(O(g(n)) \neq O(f(n)))$

Beispiele

$O(2*n-1) = O(n)$

$O(n*(n+1)/2) = O(n^2)$

$O(\mathrm{ld}\ n) = O(\log n)$

$O(\log n^2) = O(\log n)$

$O(n*\log n) < O(n^2)$

$O(\log n) < O(n^{1/2})$

Beispiel: Binäres Suchen

$$A(n) = 1 + A(n/2)$$
$$A(1) = 1$$

$$A(n) = 1 + \mathrm{ld}\, n \;\Rightarrow\; O(A(n)) = O(\log n)$$

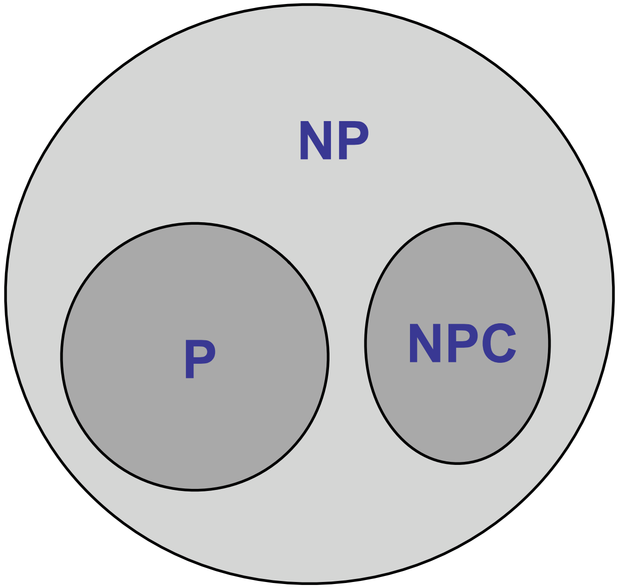
P- und NP-Probleme

P-Probleme sind mit polynomialem Aufwand lösbar
NP-Probleme sind nicht mit polynomialem Aufwand lösbar

NP steht für “nichtdeterministisch polynomial”

Ob P=NP oder P≠NP ist ist ungelöst!

$P \subseteq NP$



NP-vollständige Probleme

Alle NP-vollständigen Probleme können mit polynomialem Aufwand aufeinander abgebildet werden.

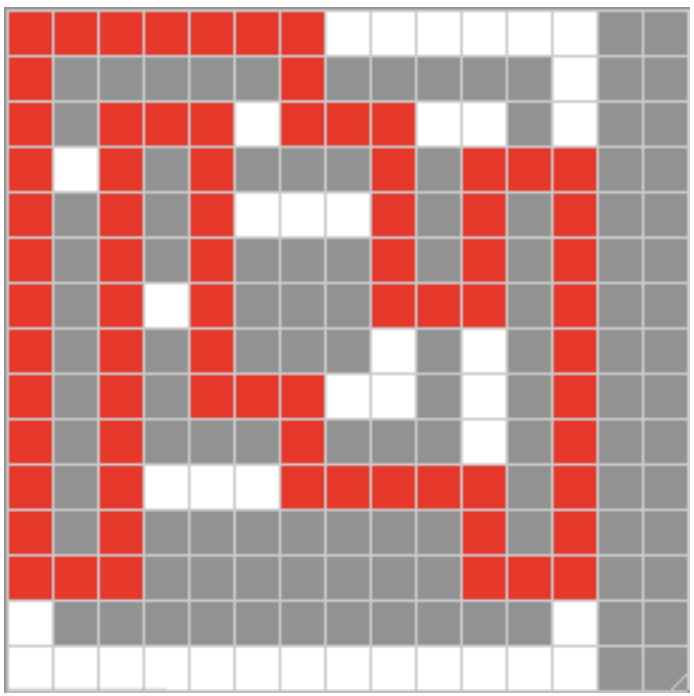
Falls ein einziges der NP-vollständigen Probleme mit polynomialem Aufwand gelöst werden kann gilt P=NP!

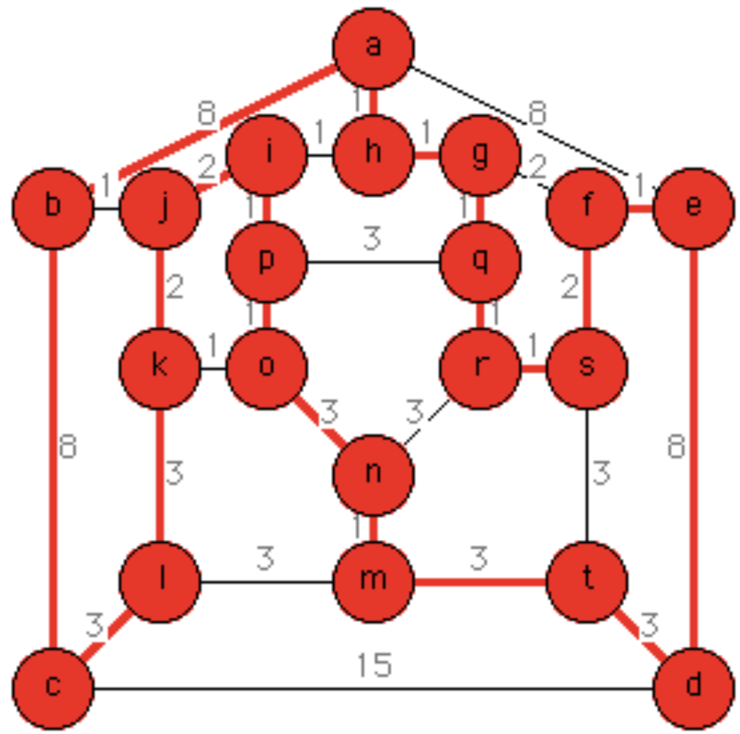
Beispiele für NP-vollständige Probleme

Traveling Salesperson
Knapsack
Scheduling
Bin-Packing
Graph Coloring
Maximal Cliques
Satisfiability

Beispiel: Travelling Salesperson (Problem des Handlungsreisenden)

Finde die kürzeste Rundreise eines Handelsreisenden, der N Städte besuchen muss (jede Stadt darf nur einmal besucht werden).





Beispiel: Knapsack (Rucksackproblem)

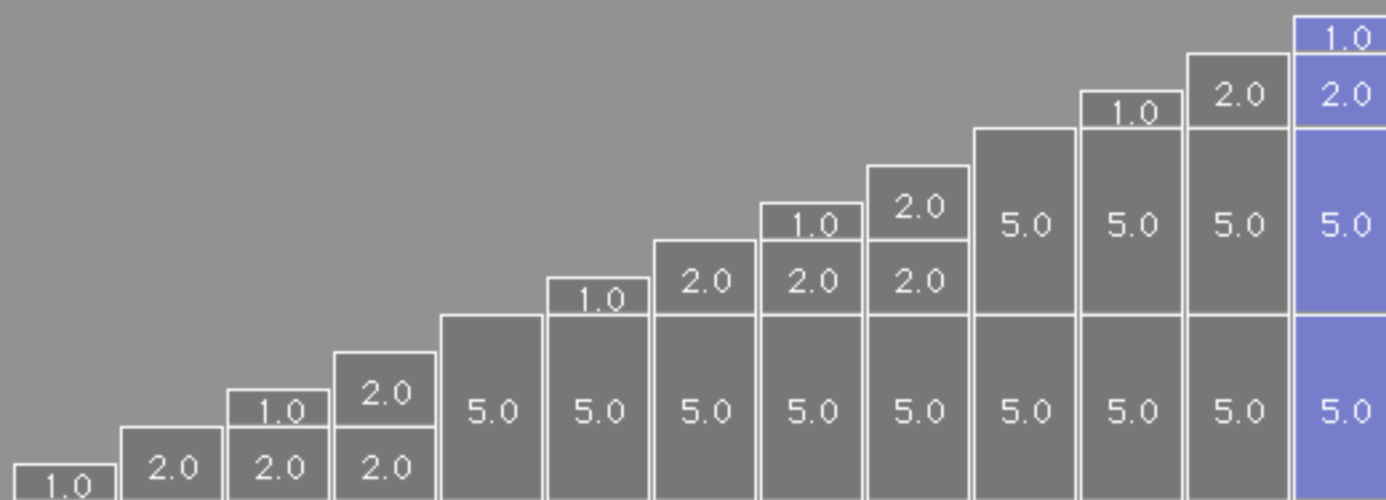
Gegeben sind Grösse (size) und Wert (profit) von N Objekten. Finde die optimale Füllung eines Rucksacks gegebener Kapazität mit einer Auswahl dieser Objekte, sodass deren Gesamtwert maximal ist. Die Objekte dürfen nicht geteilt und die Kapazität des Rucksacks darf nicht überschritten werden.

	size	profit	1	2	3	4	5	6	7	8	9	10	11	12	13
0	2	1.0	0.0	1.0	1.0	2.0	2.0	3.0	3.0	4.0	4.0	5.0	5.0	6.0	6.0
1	3	2.0	0.0	1.0	2.0	2.0	3.0	4.0	4.0	5.0	6.0	6.0	7.0	8.0	8.0
2	7	5.0	0.0	1.0	2.0	2.0	3.0	4.0	5.0	5.0	6.0	7.0	7.0	8.0	9.0

Beispiel: Subset Sum (Teilsummenproblem)

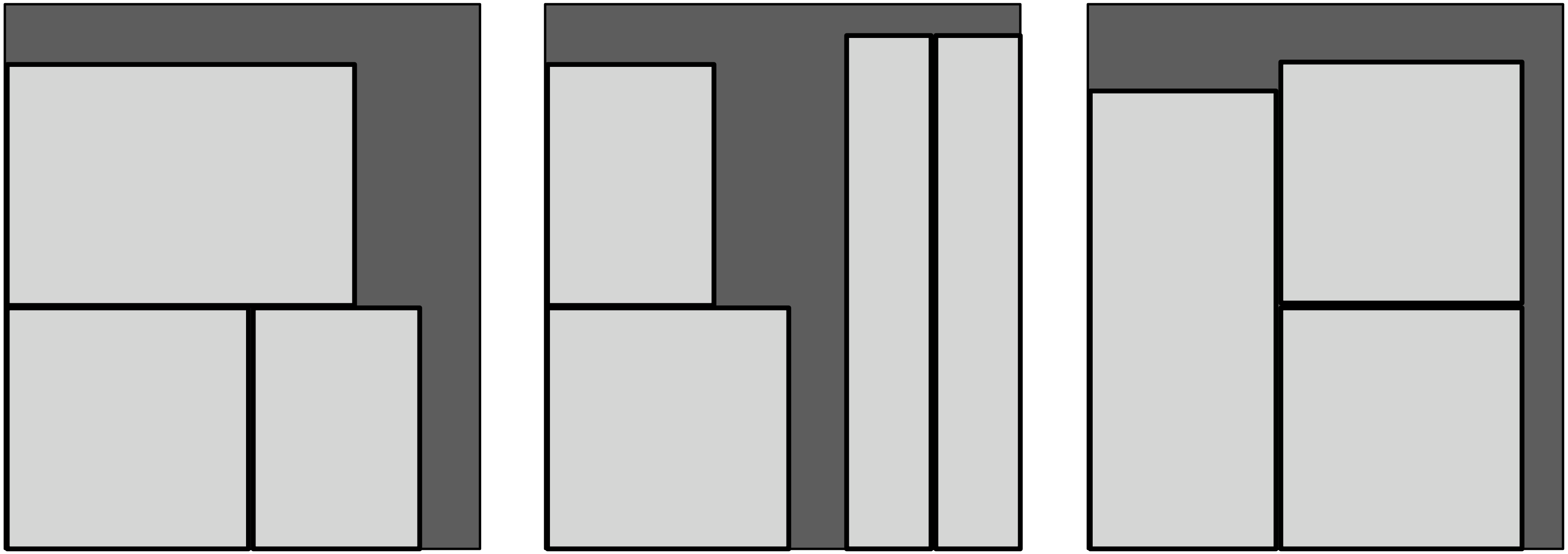
Wähle aus N gegebenen natürlichen Zahlen eine Teilmenge, sodass deren Summe gleich einer ebenfalls gegebenen natürlichen Zahl ist.

	size	profit	1	2	3	4	5	6	7	8	9	10	11	12	13
0	1	1.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0	11.0	12.0	13.0
1	2	2.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0	11.0	12.0	13.0
2	5	5.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0	11.0	12.0	13.0



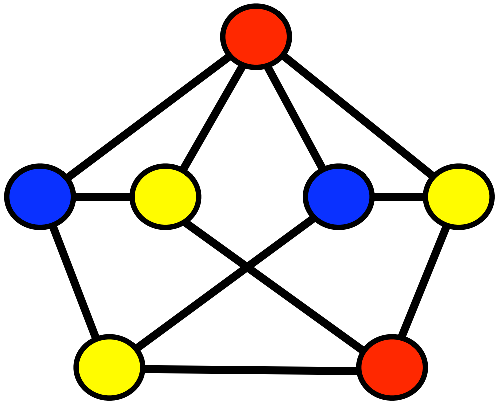
Beispiel: Bin Packing

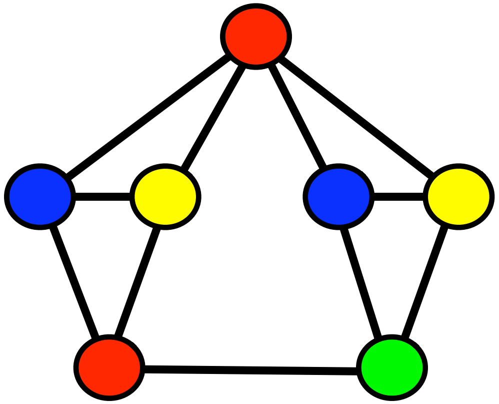
Packe N gegebene Schachteln (bins) so in Container gegebener Grösse, dass mit einer minimalen Anzahl von Containern das Auslangen gefunden wird.



Beispiel: Three Color Problem (Dreifarbenproblem)

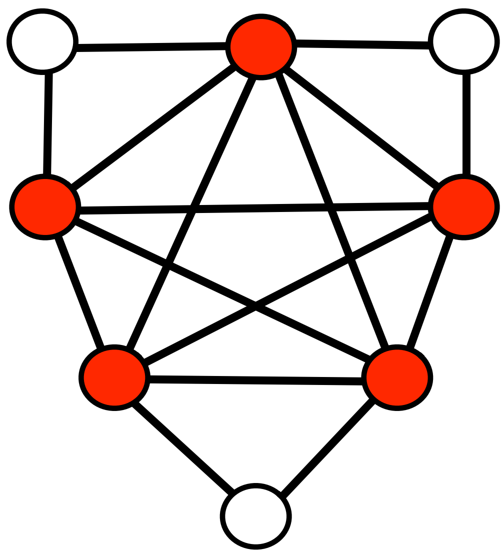
Können die N Knoten eines gegebenen Graphen so mit drei Farben gefärbt werden, dass benachbarte Knoten unterschiedlich gefärbt sind?





Beispiel: Maximal Cliques (Cliquenproblem)

Finde in einem gegebenen Graphen jene maximale Teilmenge von Knoten, die sämtliche untereinander verbunden sind. (Eine Clique ist eine Menge von Personen, von denen jede alle anderen Personen kennt.)



Nicht entscheidbare Probleme

Es gibt Aufgaben die nicht algorithmisch lösbar sind.

zB: Halteproblem: Es gibt keinen Algorithmus der entscheidet ob ein beliebiges Programm terminiert!

Indirekter Beweis

Annahme: Es gibt eine Boole'sche Funktion T(P) die genau dann den Wert true liefert wenn die Prozedur P terminiert.

Die Prozedur P kann beliebig definiert werden, zum Beispiel auch so:

procedure P {while T(P) {} }

Widerspruch: Falls P terminiert liefert T(P) true und P terminiert nicht!

Daraus folgt: Obige Annahme ist unmöglich!