Komplexität von Algorithmen

Komplexität

- Die ***Komplexität von Algorithmen*** ist ein wichtiges Teilgebiet der ***Theoretischen Informatik***.
- Es treten bei Algorithmen zwei unterscheidbare Fälle von Problemen auf
  - Berechnungsaufwand
  - Speicherplatzbedarf

Quelle u.a.:
https://de.wikipedia.org/wiki/Komplexitätstheorie

Komplexität

- Grundlegende Beispiele für diese beiden Fälle bei Suchverfahren sind
  - der Berechnungsaufwand bei der Tiefensuche mit backtracking
  - Speicherplatzbedarf bei der Breitensuche

Komplexität

- Zur Beschreibung des Aufwands und Vergleich verschiedener Algorithmen wird das Landau-Symbol verwendet, ein großes O.
- Bei kleinen Suchräumen ist der Aufwand belanglos, entscheidend wird er, wenn man ihn in Abhängigkeit vom Zuwachs der Verzweigungsbreite und Verzweigungstiefe im Suchgraphen betrachtet.

Komplexität

- Für das einfache Rucksackproblem haben wir auf jeder Stufe nur zwei Alternativen (einfüllen oder nicht einfüllen), bei n zur Verfügung stehenden Teilen können wir daher als Obergrenze für den Berechnungs-aufwand bei der Tiefensuche $\boldsymbol{O(2^n)}$ angeben.

Komplexität

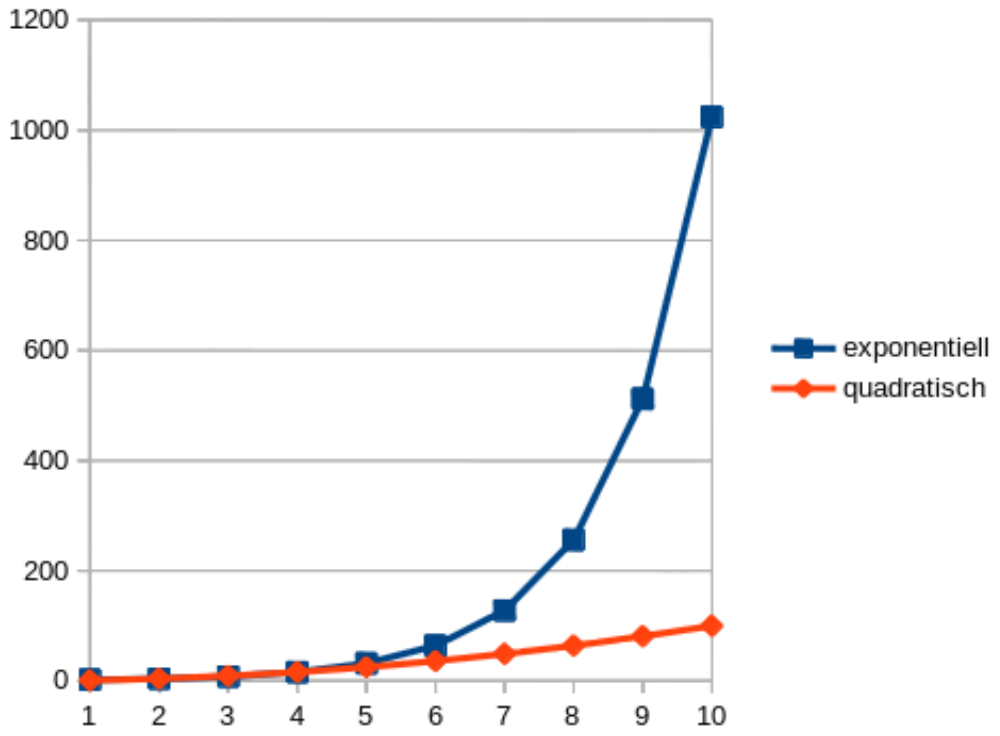
- ***O(2^n)*** ist ein sehr einfacher Ausdruck, der beschreibt, dass der Berechnungsaufwand prinzipiell proportional zur Zweierpotenz der verwendeten Stücke ansteigt.
- Einen solchen Anstieg bezeichnet man als ***exponentiell***.

Komplexität

- Einen Anstieg proportional zum Quadrat einer Größe, also ***O(n²)*** bezeichnet man als einen Fall eines ***polynomialen*** Anstiegs.
- Ein Beispiel ist die Abhängigkeit der Anzahl der Felder eines Quadrats von der Kanteneinteilung.
- ***O(n³)***, ***O(n⁴)*** … sind andere Fälle eines polynomialen Anstiegs

Komplexität

n	exponentiell	quadratisch
1	2	1
2	4	4
3	8	9
4	16	16
5	32	25
6	64	36
7	128	49
8	256	64
9	512	81
10	1024	100
11	2048	121
12	4096	144
13	8192	169
14	16384	196
15	32768	225
16	65536	256
17	131072	289
18	262144	324
19	524288	361
20	1048576	400



Komplexität

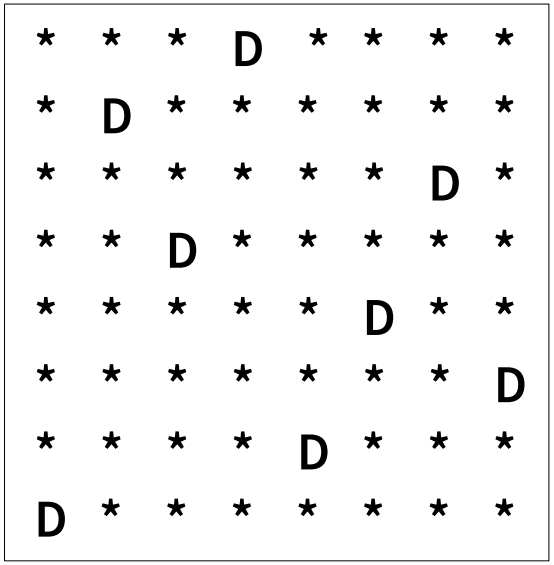
- Als einfache Grundregel kann man feststellen, dass exponentielles Wachstum dazu führt, dass ein solcher Algorithmus für große n schließlich so schnell im Aufwand wächst, dass er nicht mehr in akzeptabler Zeit zu einer Lösung führt.
- Das Problem wird dann zwar „theoretisch“ gelöst, aber nicht im Sinn der Informatik.
- „Gewünscht“ wäre prinzipiell ein polynamialer Berechnungsaufwand.

Komplexität

- Beim verallgemeinerten Problem der acht Damen lässt sich die Auswirkung sehr deutlich machen.
- Das völlig blinde Suchprogramm mit Tiefensuche arbeitet mit dem Aufwand $\boldsymbol{O((n^2)^n)}$.
- Wie in der tabellarischen Darstellung (s.o.) erkennbar wird, ist entscheidend daran der Exponent n und bereits bei kleinen n wird der Zeitaufwand bei diesem Algorithmus zu groß.

Komplexität

- Das optimierte Programm zur Tiefensuche arbeitet zwar deutlich besser, so dass auch größere n (insbesondere eben die 8) noch gelöst werden.



- Aber auch bei dem optimierten Programm gilt prinzipiell ein exponentieller Anstieg des Aufwands und bei großen n geht die praktische Lösbarkeit verloren.

Komplexität

- Das grundsätzliche Problem ist aber, dass es eine ganze Klasse von Problemen gibt, die ***n-p-vollständigen*** Probleme, zu denen es (bisher ?) keinen Algorithmus mit polynomialem Laufzeitverhalten gibt.
- Bei diesen Problemen trotzdem zu akzeptablen Optimierungen zu kommen, ist eine wichtige Aufgabe.

Komplexität

- Turau gibt in seinem Buch ***Algorithmische Graphentheorie*** im Abschnitt zu den Komplexitätsklassen **P**, **NP** und **NPC**, also *polynomial*, *nicht deterministisch polynomial* und **NP** - *vollständig* Beispiele:
- Der Algorithmus von Dijkstra hat einen Aufwand, der ihn in **NP** einordnet.

  Wir können mit polynomialem Aufwand eine (*wie auch immer*) gefundene Lösung auf Richtigkeit testen.
- Beispiele dort zu **NPC** :

Komplexität

NPC - Probleme aus dem Bereich der Graphentheorie laut Turau:

- Färbbarkeit von Graphen
- Unabhängige Mengen in Graphen
- **Hamiltonscher Kreis**
- **TSP travelling salesperson problem**
- Aufspannender Baum mit begrenztem Eckengrad

Komplexität

Hamiltonscher Kreis

- Gibt es in einem Graphen einen geschlossenen Weg, der jeden Knoten genau einmal enthält?

TSP travelling salesperson problem

- Gibt es in einem bewerteten Graphen einen Hamiltonschen Weg mit einer Länge unter einem vorgegebenen Grenzwert?
  (-> *minimaler Länge*)