Algorithmen und Komplexität

Semester Nr. 2 Algorithmen und Komplexität bei Karl Stroetmann

Inhaltsverzeichnis

**Es wurden keine Einträge für das Inhaltsverzeichnis gefunden.**

# Überblick

1. Komplexität von Algorithmen

* Groß O-Notation

1. Rekurrenz Gleichungen

* Master Theorem

1. Sortier-Algorithmen:

* Sortieren durch einfügen (Insertion Sort)
* Sortieren durch Auswahl (Selection Sort)
* Sortieren durch Mischen (Merge Sort)
* Quicksort
* Radix Sort
* Heapsort

1. Abstrakte Datentypen
2. Dictionaries (und Mengen)

* Binäre Bäume
* AVL – Bäume + 2-3-Bäume
* Hash Tabellen
* Tries (Spezialfall Strings)

1. Prioritäts Warteschlangen
2. Graphentheoretische Algorithmen

Man muss nun zwischen einem Algorithmus und einem Programm unterscheiden. Ein Algorithmus ist eine abstrakte Darstellung, wie ein gegebenes Problem gelöst werden kann. Ein Programm hingegen ist die konkrete Implementierung dessen.

Des Weiteren sollte ein Algorithmus drei Kriterien erfüllen:

1. Ein Algorithmus muss korrekt sein
2. Ein Algorithmus sollte effizient sein in Bezug auf die Rechenzeit und den Speicherverbrauch
3. Ein Algorithmus sollte einfach sein.

# Groß-O-Notation

## Motivation

Wie berechnen Rechner die Zeiten eines Algorithmus?

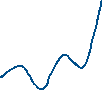
1. Implementierung in Programmiersprache
2. Zählen von arithmetischen Operationen und Speicherzugriffen
3. Nachschlagen der Zeit der Operationen im Prozessorhandbuch
4. Berechnung der Rechenzeit

Ein Bild, das Karte enthält.

Automatisch generierte BeschreibungDie Groß-O-Notation ist eine abstrakte Möglichkeit, die das Wachstum der Rechenzeit in Abhängigkeit von der Größe der Eingabe beschreiben. Die O-Notation soll von konstanten Faktoren und unwesentlichen Termen abstrahieren.



Man definiert einen X-Wert (k), ab dem die Funktion  (rot) immer unter  liegt. Das c muss ebenfalls definiert werden und gibt einen Faktor der Funktion g(x) (blau) an, für welche dann das Wachstum von  ab k immer unterhalb von  verläuft.



k



### Definition der Groß-O-Notation



## Aussagen der Groß-O-Notation

### Reflexivität der Groß-O-Notation

Für alle Funktionen  gilt: 

Beweis: Seien  und  definiert, dann folgt die Behauptung direkt aus der Ungleichung:



### Multiplikation von Konstanten

Angenommen man hat die Funktionen  und eine Zahl . Dann hat man auch



Beweis: Die Prämisse  impliziert, dass es die Konstanten  und  gibt, so dass

 gilt.

Nun wird die Ungleichung mit d multiplizieren:



Nun werden  und  definiert:



Per Definition impliziert dies:



Erinnerung: Die Vorherige Aussage zeigt, dass die Groß-O-Notation von konstanten Faktoren abstrahiert.

### Addition

Angenommen es seien  gegeben. Dann hat man auch



Beweis: Die Vorbedingung  und  impliziert, dass es die Konstanten