Algorithmen und Komplexität

Semester Nr. 2 Algorithmen und Komplexität bei Karl Stroetmann

Inhaltsverzeichnis

[Überblick 3](#_Toc42204192)

[Groß-O-Notation 4](#_Toc42204193)

[Motivation 4](#_Toc42204194)

[Definition der Groß-O-Notation 4](#_Toc42204195)

[Aussagen der Groß-O-Notation 4](#_Toc42204196)

[Reflexivität der Groß-O-Notation 4](#_Toc42204197)

[Multiplikation von Konstanten 4](#_Toc42204198)

[Addition 5](#_Toc42204199)

[Transitivität der Groß-O-Notation 5](#_Toc42204200)

[Grenzwert der Groß-O-Notation 6](#_Toc42204201)

[Lösen einer Rekurrenzgleichung 6](#_Toc42204202)

[Ausnahmefälle 6](#_Toc42204203)

[Arten der Groß-O-Notation 6](#_Toc42204204)

[Master-Theorem 7](#_Toc42204205)

[Bedingungen für das Master-Theorem 7](#_Toc42204206)

[Anwendung des Master-Theorems 7](#_Toc42204207)

# 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

Ein Bild, das Screenshot, Vogel enthält.

Automatisch generierte Beschreibung

### Multiplikation von Konstanten

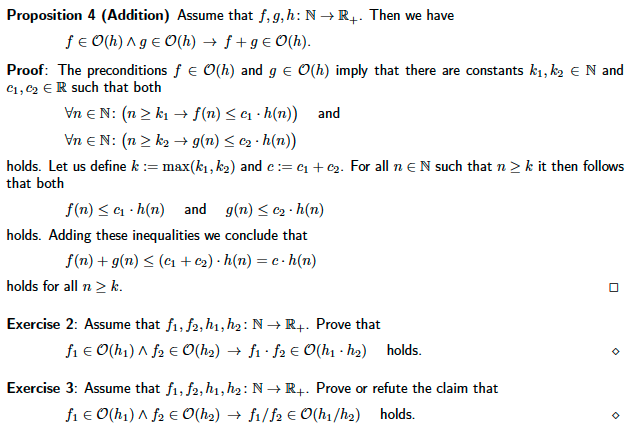
Ein Bild, das Screenshot, Vogel enthält.

Automatisch generierte Beschreibung

Ein Bild, das Vogel enthält.

Automatisch generierte Beschreibung

### Addition

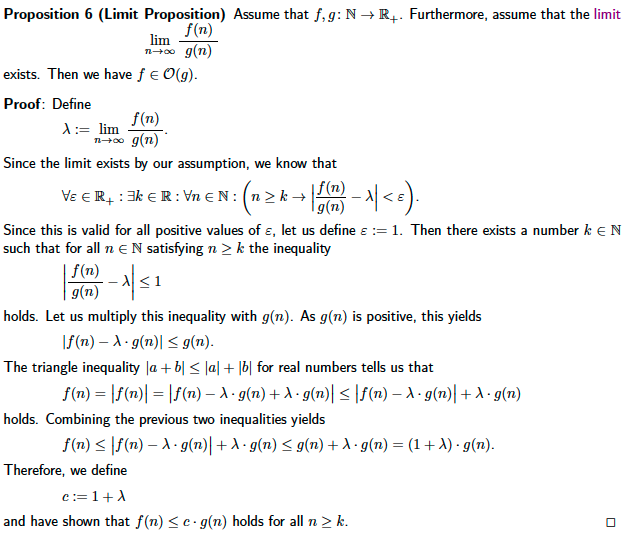


### Transitivität der Groß-O-Notation

Ein Bild, das Tisch, Anzeige, viele, Vogel enthält.

Automatisch generierte Beschreibung

### Grenzwert der Groß-O-Notation



## Lösen einer Rekurrenzgleichung

1. Homogenen Teil lösen ()
   * Die Rekurrenzgleichung ohne +1 am Ende
2. Inhomogenen Teil lösen ()
   * Die gesamte Rekurrenzgleichung
3. Einsetzen in die allgemeine Lösung
   * 
4. Gleichungssystem mit den Anfangsbedingungen Aufstellen
5. Lösen des Gleichungssystems und  &  bestimmen
6. Allgemeine Lösung aufstellen
   * 

### Ausnahmefälle

1. Gilt , dann ist die allgemeine Lösung
   * 
2. Wenn , dann ist der Ansatz zur Lösung der inhomogenen Rekurrenzgleichung
   * 

## Arten der Groß-O-Notation

1. Die O-Notation:
   * 
2. Die -Notation:
   * 
3. Die -Notation:
   * 

# Master-Theorem

Mit dem Master-Theorem kann man einfacher die Komplexität einer rekursiven Funktion bestimmen.

## Bedingungen für das Master-Theorem

1. , so dass , so dass  und
2. die Funktion  erfüllt die Rekurrenzrelation



## Anwendung des Master-Theorems

Man kann durch das Master-Theorem alle benötigten Variablen schnell ablesen und so die Komplexität des Algorithmus in einen der folgenden Fälle einordnen.

1. 
2. 
3. 