Algorithmen und Datenstrukturen

Christian Rieck, Arne Schmidt

Übung #7

01.02.2018

Übersicht

- Besprechung Blatt 5
- Klausurhinweise
- Klausuraufgaben

Aufgabe 4: Master-Theorem II

Worauf man im Allgemeinen achten sollte:

- Sind alle α_i’s konstant?
- Sind alle α_i’s zwischen 0 und 1?
- Gibt es ein konstantes $k \in \mathbb{R}$?
- Nur ganzzahlige rekursive Aufrufe?
- Keine negativen rekursive Aufrufe?

Rückgabe Blatt 5: Ab nächste Woche Dienstag/Mittwoch (genauer per Mail)

Bekanntgabe, wer **Studienleistung** erbracht hat: Selber Zeitraum

Fragen?

Klausur

- Findet statt
  - Am 23.02.2018
  - Zwischen 11 und 14 Uhr
    - (Genaue Zeit wird noch bekanntgegeben)
  - Zeit: 120 Minuten
- Raumaufteilung wird kurz vor der Klausur (2-3 Tage vorher) bekanntgegeben.
- Mitbringen:
  - Studiausweis (+ Personalausweis, falls kein Bild)
  - Stift
  - Wörterbuch (falls nötig)
- Papier gibt es von uns!
- Klausurergebnisse vorraussichtlich 28.02.18

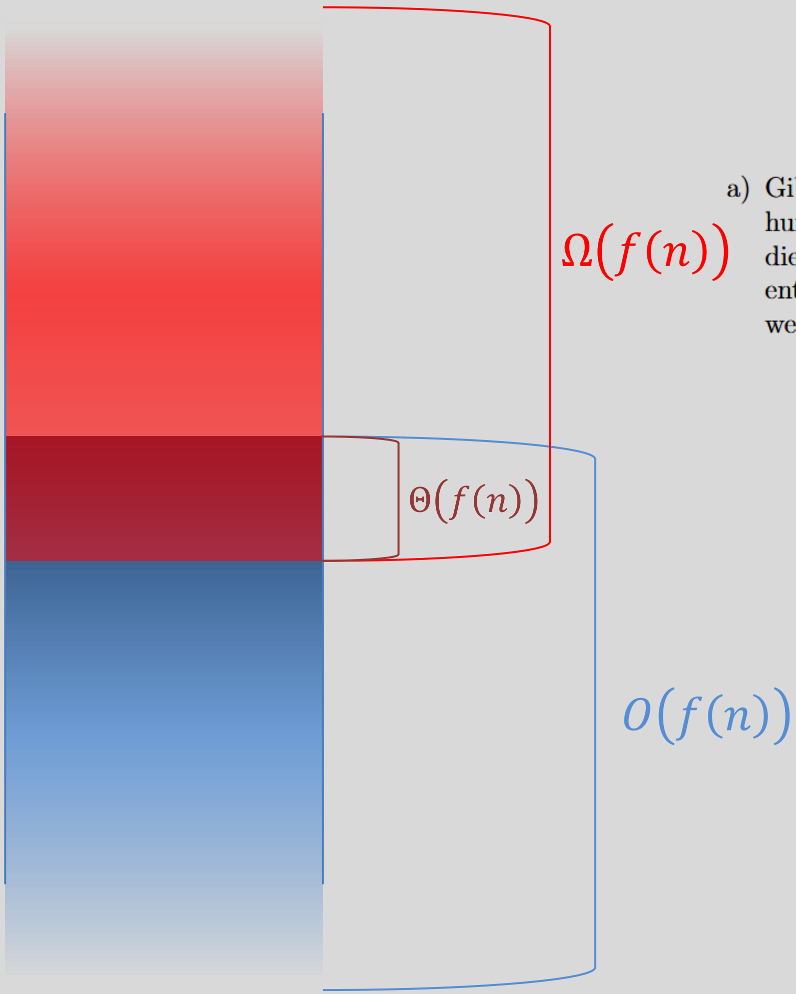
Klausur – Inhalt

Alte Frage: Was soll man lernen?

- Einfache Antwort: alles!
  - Algorithmen:
    - BFS/DFS
    - Fleury
    - Binäre Bäume, AVL-Bäume
      - Insert, Delete, Restructure,…
    - Quicksort, Bubblesort, Mergesort, Radixsort,…
    - Median
    - …
  - Definitionen/Sätze:
    - Pfade, Wege, Eulerweg/-tour, Hamiltonpfad/-kreis,…
    - Wann ist Graph eulersch?
    - Binäre Bäume
      - Voll, vollständig, AVL-Eigenschaften,…
    - Master-Theorem
    - Laufzeiten von Algorithmen
    - ...

Eine Klausuraufgabe!

Laufzeiten



a) Gib für die folgenden Paare von Funktionsklassen an, in welcher Teilmengenbeziehung sie zueinander stehen. Schreibe dazu in die mittlere Spalte „=“, falls A genau dieselben Funktionen enthält wie B, „$\subset$“ wenn alle Funktionen aus A auch in B enthalten sind, „$\supset$“ wenn alle Funktionen aus B auch in A enthalten sind, und „$\times$“, wenn dies alles nicht zutrifft. Eine Begründung ist *nicht* erforderlich.

A	Antwort	B
$\mathrm{O}(n)$		$\mathrm{O}(n - \log n)$
$\mathrm{O}(n)$		$\Theta(n)$
$\Omega(n^2)$		$\Omega(n \log n)$
$\mathrm{O}(1)$		$\mathrm{O}(2 + \cos(\pi n))$
$\Omega(5^n)$		$\Omega(4^n)$
$\mathrm{O}(\log(n^2))$		$\mathrm{O}((\log(n))^2)$
$\mathrm{O}(1)$		$\Omega(1/n)$
$\Theta(\sqrt{n})$		$\Theta(\log n)$

Mediane

Rang-k Element m in X:

$$|\{x \in X: x \leq m\}| \geq k$$

$$|\{x \in X: x \geq m\}| \geq n - k$$

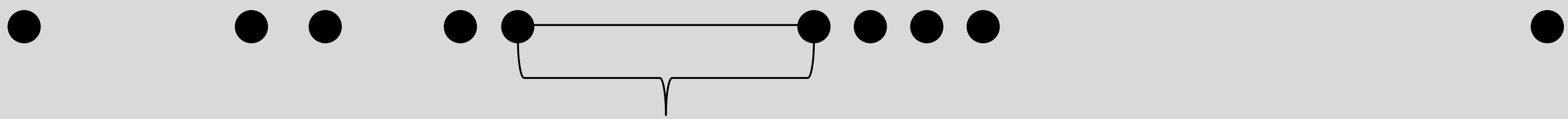
Für einen Median m in X gilt:

$$|\{x \in X: x < m\}| \leq \frac{n}{2}$$

$$|\{x \in X: x > m\}| \leq \frac{n}{2}$$

m Eindeutig bei $|X|$ ungerade

Mediane



Jeder Punkt in diesem
Bereich ist ein Median!

Bei $X = \{1,2,3,4,5,6,7,8\}$ sind sowohl 4 als auch 5 ein Median.

Wie finden wir nun den Median?

Median – Algorithmus

Idee: Führe Partition von Quicksort aus und schaue in welchem Teil der Median liegen muss.

Problem: Laufzeit im Worst-Case $O(n^2)$

Wurzel des Übels: Pivotisierung!

Median – Pivotisierung

Idee:

1. Teile Zahlen in 5er Gruppen
2. Bestimme Median in jeder Gruppe
3. Bestimme Median der Mediane m
4. Benutze m als Pivotelement

Wieso hilft das?

Median – Pivotisierung II

Sortiere gedanklich 5er Gruppen bzgl. deren Median

$\leq m$	$\leq m$	$\leq m$	$\leq m$	$\leq m$					
$\leq m$	$\leq m$	$\leq m$	$\leq m$	$\leq m$					
$\leq m$	$\leq m$	$\leq m$	$\leq m$	m	$\geq m$	$\geq m$	$\geq m$	$\geq m$	$\geq m$

Etwa mindestens $\frac{1}{4}$ der Elemente (rot) können nicht rechts vom Pivot liegen

Etwa mindestens $\frac{1}{4}$ der Elemente können nicht links vom Pivot liegen

Daher: Rekursiver Aufruf auf etwa $\frac{3}{4}n$ Elemente!

Median – Laufzeit

Wir haben also als Laufzeit:

$$T(n) = \underbrace{T\left(\frac{1}{5}n\right)}_{\substack{\text{Bestimmung Median} \\ \text{der Mediane}}} + \underbrace{T\left(\frac{3}{4}n\right)}_{\substack{\text{Rekursion in} \\ \text{linke/rechte Teilhälfte}}} + \underbrace{\Theta(n)}_{\substack{\text{Aufteilen in Gruppen/} \\ \text{Pivotisieren/} \\ \text{…}}}$$

Mit dem Master-Theorem folgt: $T(n) \in \Theta(n)$