

Algorithmen I - Sommersemester 2014

Tutorium Nr. 1

Tobias Hornberger | 24. April 2014

INSTITUT FÜR THEORETISCHE INFORMATIK



Tobias Hornberger

- 4. Semester Informatik
- Algorithmen I vor 2 Semestern gehört
- erstes Mal Tutor

Vorschläge/Anmerkungen/Feedback ist also sehr erwünscht

- Email: `saibot1207@googlemail.com`
- Github: `https://github.com/Strandtasche`
- Bei Fragen per Email: Antwort an Alle Email mit Betreff „Algo1Tut“ an mich schicken.

Vorstellungsrunde

- Das Tutorium ist nicht dazu gedacht Übung/Vorlesung zu ersetzen, sondern als Ergänzung.
- Der Besuch wird empfohlen
- Keine Frontalveranstaltung
 - Wenn Fragen auftauchen: einfach Fragen
 - Es gibt keine dummen Fragen

Speziell für dieses Semester:

- Viele Feiertage fallen auf Donnerstage
- in diesen Wochen bitte andere Tutorien besuchen

- Vorlesungshomepage:
<http://algo2.iti.kit.edu/AlgorithmenI2014.php>
 - Interessante Organisatorische Details
 - Vorlesungsfolien
 - Die Übungsblätter
 - Prof. Sanders Buch
 - Literaturliste
- Meine Folien:
<https://github.com/Strandtasche/AlgoTutSS2014>
- ILIAS: <https://ilias.studium.kit.edu>
- Fakultät für Informatik → SS 2014 → Algorithmen I mit Übung

Das meiste wurde schon in der Vorlesung gesagt:

- 9 Tage Bearbeitungszeit: von Mittwoch bis Freitag der nächsten Woche
- Abgabe 12:45 Uhr im Kasten hier vor dem Raum (-118)
- 2er-Teams sind erlaubt
- Teamwechsel im Semester sind nicht vorgesehen, können aber in Sonderfälle mit Rücksprache durchgeführt werden
- Wichtig Wichtig: Tutoriumsnummer rechts oben groß auf die Abgabe!
- Kein Verpflichtender Übungsschein sondern Klausurbonus von 1/2/3 Punkten
- Zum ersten Mal dieses Jahr: Programmieraufgaben
 - 4 Stück, über das Semester verteilt, in Java
 - keine Klassenstrukturen sondern tatsächlich nur der Algorithmus
 - Praktomat Abgabe: Ohne Verpflichtenden Code Style
 - Dazu mehr wenn die erste Aufgabe kommt...



Beispielcode:

a ist ein Array von \mathbb{N}

```
1  int max = -1
2  int maxLoc
3  for i in range(0, |a|-1):
4      if max < a[i]:
5          max = a[i]
6          maxLoc = i
```

Pseudocode:

```
1  function(int a[], int key)
2  int indexMin = 0
3  int indexMax = |a| - 1
4  int indexMid
5
6  while indexMax >= indexMin
7      indexMid = (indexMin + indexMax) / 2
8
9      if a[indexMid] = key
10         return indexMid
11     else if a[indexMid] < key
12         indexMin = indexMid + 1
13     else
14         indexMax = indexMid - 1
15
16  return -1
```

Bekannt aus GBI:

zum Untersuchen von Laufzeitverhalten bei Eingaben verschiedener Größen

- **Best Case:** Cool zu wissen, aber meistens irrelevant
- **Average Case:** Sehr interessant, aber unhandlich/schwer zu berechnen
- **Worst Case:** Hiermit beschäftigen wir uns

Bezug

Begriffe beziehen sich auf deterministische Algorithmen mit fester Eingabegröße.

Formeln

- $\mathcal{O}(f(n)) = \{ g(n) : \exists c > 0 : \exists n_0 \in \mathbb{N}_+ : n \geq n_0 : g(n) \leq c * f(n) \}$
- $\Omega(f(n)) = \{ g(n) : \exists c > 0 : \exists n_0 \in \mathbb{N}_+ : n \geq n_0 : g(n) \geq c * f(n) \}$
- $\Theta(f(n)) = \mathcal{O}(f(n)) \cap \Omega(f(n))$
- $o(f(n)) = \{ g(n) : \lim_{n \rightarrow \infty} \frac{g(n)}{f(n)} = 0 \}$
- $\omega(f(n)) = \{ g(n) : \lim_{n \rightarrow \infty} \frac{g(n)}{f(n)} = \infty \}$

Bemerke:

$\log(n)$ ist immer ohne Basis angegeben. Wieso?

Basisumrechnung

$$\log_b(n) = \frac{\log_d(n)}{\log_d(d)}$$

\Rightarrow Ist das gleiche wie $\frac{1}{\log_d(d)} * \log_d n$

■ $14n^3 + 3n^2 + 7n + 15$

■ $8n + 7$

■ $15n^2 + 3$

■ 1500

■ -7

■ $n!$

■ n^n

■ $\log n$

■ $\frac{1}{2}n^2$

■ $n + n * \log n$

■ $4n + 15n$

Zeigen Sie, dass

- $\mathcal{O}(\log n) = \mathcal{O}(\log n^2)$
- $f(n) = n! \in \mathcal{O}(n^n)$ ist

A ist ein Array aus \mathbb{N} , x ist ein int

```
1. for i in range(0, |A|-1):  
2.     if A[i] = x  
3.         return True  
4. return False
```

Was tut es?

Laufzeit?

Schleifeninvariante?

Bis zum nächsten Mal

