

Informatik II: Algorithmen und Datenstrukturen SS 2017

Vorlesung 13b, Mittwoch, 26. Juli 2017
(Evaluation, Klausur, Aktuelle Forschung)

Prof. Dr. Hannah Bast
Lehrstuhl für Algorithmen und Datenstrukturen
Institut für Informatik
Universität Freiburg

Blick über die Vorlesung heute

■ Inhalt

- Evaluationsergebnisse Zusammenfassung + Ausblick
- Klausur Termin + Modus + Aufgaben
- Vorstellung Lehrstuhl Aktuelle Forschung + nächste VLen

■ Teilnahme

- Noch aktive Teilnehmer*innen : **160**
- An der Evaluation teilgenommen : **121**
 - 92** x Info, **8** x Info Nebenfach, **20** x ESE, **18** x Sonstige
 - 70** x 2. Semester, **36** x 4. Semester, **31** x höhere Semester
- Nominierungen für Lehrpreis : **83**
- Im Folgenden, eine Zusammenfassung des Feedbacks

Die vollständigen Ergebnisse, inklusive aller Statistiken und aller Freitextkommentare, finden Sie auf dem Wiki

■ Art und Weise

- **Viel gelernt:** 67% trifft voll zu, 26% trifft zu, 6% ok
- **Verständlich:** 83% trifft voll zu, 12% trifft zu, 5% ok
- **Niveau:** 57% angemessen, 37% hoch, 6% tief
- **Qualität:** 72% sehr gut, 22% gut, 6% geht so
- Dozentin: motivierend, verständlich, kompetent, lockere Stimmung, unterhaltsam, humorvoll, authentisch, engagiert
- Inhalt: ein Thema pro Woche, gute Mischung aus Theorie und Praxis, "keine Overkill-Formalismen", Bezüge zu anderen Themen / Fragen über den Tellerrand
- Tempo, Bilder, Live-Coding: nächste Folie

■ Tempo, Bilder, Live-Coding

- Viele fanden das Tempo gerade richtig, einigen wenigen war es zu langsam bzw. hätten sich mehr Stoff gewünscht

"Sehr angenehmes Tempo ohne Gehetze"

"Schneller durch den Stoff gehen, sind nicht in der Schule"

- Viele fanden das gemeinsame Entwickeln von Zeichnungen und Code sehr hilfreich, aber einigen dauert es zu lange

"Das Live-Coding hat Klasse und lockert die VL ungemein auf ... bitte beibehalten"

"Weniger Zeit für das Zeichnen von simplen Schaubildern, stattdessen tiefergehende theoretische Hintergründe bzw. ausführlichere mathematische Beweis"

■ Übungsblätter

- Aufwand relativ zu ECTS ... 1 = sehr hoch, 5 = sehr gering

5% x 1 40% x 2 50% x 3 4% x 4 0% x 5 diese Veranstaltung

14% x 1 30% x 2 52% x 3 3% x 4 1% x 5 Durchschnitt Informatik

- Aufgabe haben viel Spaß gemacht, interessant und realitätsnah, gut machbar, sehr gut auf die Vorlesung abgestimmt, schnelle Korrektur, Musterlösungen, 3 Sprachen zur Auswahl
- Immer noch vereinzelt Wunsch nach persönlichem Tutorat
- Lob für die Tutoren: Maya Schöchlin, Sebastian Holler, Daniel Tischer, Daniel Bindemann, Danny Stoll, Simon Selg
- Lob für die Arbeit des Assistenten: Axel Lehmann (SVN)

■ Materialien / Online Support

- **Hilfreich:** 81% trifft voll zu, 10% trifft zu, 9% ok
- Konsumiert durch Anwesenheit / Aufzeichnungen / Folien:
 - 23% Anw, 36% Aufz, 30% beides, 11% Folien **diese Veranstaltung**
 - 36% Anw, 16% Aufz, 18% beides, 30% Folien **Durchschnitt Informatik**
- Für die Aufzeichnungen und den Schnitt großen Dank an:
Frank Dal-Ri (Technik) & Alexander Monneret (Schnitt)
- Sehr aktives + hilfsbereites Forum, ausgezeichnetes Team, Deluxe-Videoaufzeichnungen, diesmal sogar Live-Stream
- Suche im Forum könnte besser sein → **stimmt**

■ Kritik / Wünsche **von letztem Mal** (SS 2015)

- Ein Tutorat gelegentlich bzw. am Anfang

Maßnahme: Fragestunde in Vorlesung 3b, früh darauf hingewiesen, dass persönliches Treffen mit Tutor möglich

- Zu viel Zeit für Malen / Farbauswahl ... eine Mehrheit fand das aber im Gegenteil gerade gut

Maßnahme: Zeichnungen teilweise vorbereitet, auch Programm nicht mehr immer "from scratch" geschrieben

- In der Mitte des Semesters teilweise zu leicht

Maßnahme: Aufgaben angepasst und auf gleichmäßiges Niveau geachtet + diverse Zusatz- bzw. Bonusaufgaben

- Was wir besser / weiter gut gemacht haben
 - Maßnahmen von der vorherigen Folie
 - Drei Programmiersprachen zur Auswahl (Python, Java, C++), mit gleichwertigen Vorlagen für die Ü-Blätter
 - Mehr "freiere" Übungsblätter, wo man sich selber einen Algorithmus überlegen / selber weiterdenken muss
 - Zusatz- bzw. Bonusaufgaben für die Unterforderten
 - Zeitmanagement ... ziemlich gut dieses Semester
 - Zahlreiche (meistens kleine) Fehler korrigiert + Folien teilweise besser strukturiert oder weiter perfektioniert
 - Etwas härterer Umgang mit Plagiaten

- Geplante Verbesserungen für nächstes Mal
 - Ein **Präsenztutorat** in der zweiten oder dritten Woche
 - Von Beginn sehr klar definieren + aufschreiben, was ein **Plagiat** ist und was nicht und was der Sinn der Regel ist
 - **LaTeX-Vorlage** für die Abgabe der Übungsblätter
 - Überlegen, wie man die (sehr wenigen) Unterforderten mehr fordern kann ohne die anderen abzuhängen
 - Aber nicht ganz sicher, ob diese wirklich unterfordert sind
 - Folien überarbeiten + verbleibende Fehler ausmerzen
 - Ich mache mir zu jeder Vorlesung ausführliche Notizen
 - Interaktivere Formate ausprobieren, zumindest ab und zu

■ Termin + Punkte

– Am **29. August 2017** von **14 – 17 Uhr**, Gebäude 101

Wir teilen Ihnen noch mit, wer wo sitzt (Forum + Mail)

– **6** Aufgaben a **20** Punkte, wir zählen die besten **5**

– Also maximal **100** Punkte

■ Endnote

– Ergibt sich linear aus der Punktzahl in der Klausur

50 – 54: **4.0**; 55 – 59: **3.7**; 60 – 64: **3.3**

65 – 69: **3.0**; 70 – 74: **2.7**; 75 – 79: **2.3**

80 – 84: **2.0**; 85 – 89: **1.7**; 90 – 94: **1.3**

95 – 100: **1.0**

■ Modus

- Die Klausur ist **open book** : sie dürfen Bücher, Papier, usw. im Gesamtgewicht bis zu **527 kg** mitbringen

Aber bitte sparsam beim Ausdrucken der Folien sein !

- Elektronische Geräte jeder Art sind nicht gestattet
- **Außerdem bitte mitbringen:**
Studenausweis, Buntstifte, Gehirn



■ Drei Typen von Aufgaben

- **Typ 1:** Einen Algorithmus, oder eine Variante davon, an einem Beispiel nachvollziehen

Siehe Buntstifte, aber **bitte kein Rot verwenden**

- **Typ 2:** Kleineres Programm schreiben oder gegebenes Programm verstehen und/oder die Laufzeit analysieren
- **Typ 3:** Kleinere Rechenaufgaben oder Beweise oder Denkaufgaben ... **siehe Gehirn**
- Auf den nächsten drei Folien ein Beispiel zu jedem Typ

Auf dem Wiki finden Sie die Klausuren vom SS 2015 und SS 2013 + drei ältere Klausuren zu AlgoDat für ESE

- Beispielaufgabe vom Typ 1 (Algorithmus am Beispiel)
 - Klausur SS 2017, Aufgabe 3.1: Zeichnen Sie den Zustand eines binären Heaps nach Einfügen von 5, 4, 3, 2, 1

OPERATION

insert(5)

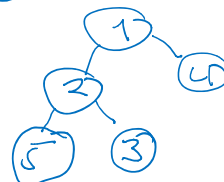
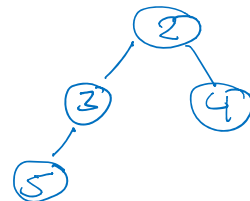
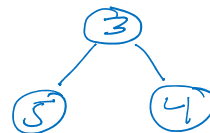
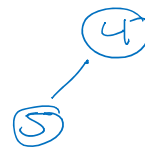
insert(4)

insert(3)

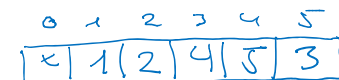
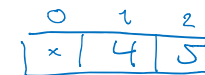
insert(2)

insert(1)

BAUM



FELD



■ Beispielaufgabe vom Typ 2 (Programm + Laufzeit)

- Klausur SS 2017, Aufgabe 3.2: Funktion, die für einen (als Feld) gegebenen Heap überprüft, ob Heapeigenschaft gilt
- Klausur SS 2017, Aufgabe 3.3: Die Laufzeit der Funktion

```
def checkHeapProperty(heap):  
    for i in range(2, len(heap)):  
        child_value = heap[i]  
        parent_value = heap[int(i/2)]  
        if child_value < parent_value:  
            return False  
  
    return True
```

n = Größe von dem Feld
Schleife läuft $n-2$ mal, Zustand viele Operationen
no Durchlauf
 $\rightarrow \Theta(n)$

4 8 7 2 9 1
 pivot \rightarrow
 TEIL 1: 2 1
 TEIL 2: 8 7 9

■ Beispielaufgabe vom Typ 3 (Rechenaufgabe / Beweis)

- Klausur SS 2017, Aufgabe 6.3: Wahrscheinlichkeit, dass bei zufälligen Pivot bei Quicksort beide Teile $\geq n/4$

Dabei: n = Feldgröße, n durch 4 teilbar, alle Zahlen sind verschieden, der Pivot gehört zu keinem der beiden Teile

Beispiel $n = 8$:
 1 2 3 4 5 6 7 8
 $\uparrow \uparrow \uparrow \uparrow \uparrow \uparrow \uparrow \uparrow$
 schlecht gut schlecht
 (beide Teile $\geq n/4$)
 $n/4 = 2$

Elemente in sortierter Reihenfolge: $x_1 < x_2 < \dots < x_n$

Pivot aus $x_1, \dots, x_{n/4}$: Teil 1 $< \frac{n}{4}$

Pivot aus $x_{n-\frac{n}{4}+1}, \dots, x_n$: Teil 2 $< \frac{n}{4}$

Sonst: beide Teile $\geq \frac{n}{4}$

$$\Rightarrow W\text{-Teil} = \frac{1}{2} \quad \square$$

■ Wie wir arbeiten

- Wir lösen praktisch relevante Probleme

Routenplanung auf Google Maps, "Search As You Type",
Semantische Suche, "Question Answering"

- Wir machen unsere Software + Ergebnisse verfügbar

Dazu braucht man gute Software, gute Dokumentation,
gute Benutzerschnittstellen, usw.

- Theorie als Werkzeug, nicht um der Theorie willen

Aber wichtig: ohne theoretisches Verständnis beim Lösen
komplexer Problem nur "Rumgehacke" und "Raterei"

■ Betreuung

- Ähnlich wie in der Vorlesung

Sehr gute Infrastruktur und Support

Abgesehen davon sollten Sie sehr unabhängig arbeiten

Insbesondere haben Sie bei uns viel Freiheit

Gut geeignet für enthusiastische Leute, die gerne praktische Probleme lösen und Sachen gemacht kriegen wollen

■ Maschinelles Lernen

- Wir benutzen zunehmend maschinelles Lernen zur Lösung vieler (nicht aller) unserer Probleme

Nicht weil es in Mode ist, sondern weil es praktisch ist

Es ist ziemlich offensichtlich, dass maschinelles Lernen der Ansatz der Wahl für komplexe Probleme ist, wie z.B. die Verarbeitung / das Verständnis von natürlicher Sprache

- Beim maschinellen Lernen geht es weniger um ausgefeilte Algorithmen sondern um ein tiefes Verständnis der grundlegenden Prinzipien und wie und warum sie funktionieren

Ein Verständnis der Algorithmen und vor allem die "Denke" aus der Info II Vorlesung ist dabei Grundvoraussetzung

■ Aktuelle Projekte und Demos

- Routenplanung (Teil von Google Maps) [demo](#)
- Visualisierung des weltweiten ÖPNV (Travic) [demo](#)
- Automatisches Malen von ÖPNV-Karten (Loom) [demo](#)
- Interaktive Semantische Suche (Broccoli) [demo](#)
- Large-Scale SPARQL+Text Suche (QLever) [demo](#)
- Question Answering (Aqqu) [demo](#)
- Question Completion [demo](#)
- Text extraction from PDF (Icecite) [paper](#)

■ Vorlesungen

- **WS 17/18: Information Retrieval** (Spezialvorlesung)

Alles was man braucht, um eine Suchmaschine (oder verwandte Services) gemäß dem Stand der Kunst zu bauen

Potpourri aus vielen Techniken und Gebieten: Algorithmen, Kodierungstheorie, Web Apps, Maschinelles Lernen, ...

- **SS 2018: Programmieren in C++** (Grundvorlesung)

C/C++ lernen von Grund auf

Inklusive Makefiles, Compiler, Linker, Debugger, Templates, und das ganze Drumherum

■ Projekte + Abschlussarbeiten

- Auf unserem Wiki:

[Liste von Themen + Informationen zum Ablauf](#)

Keine festen Zeiten, einfach nachfragen, siehe Folie 18

Die Wiki-Seite vorher gründlich durchlesen

[Leitfaden zum Schreiben einer Abschlussarbeit](#)