

Algorithmen und Datenstrukturen (ESE)
Entwurf, Analyse und Umsetzung von
Algorithmen (IEMS)
WS 2012 / 2013

Vorlesung 15, Dienstag, 12. Februar 2013
(Evaluation, Klausur, Vorstellung Lehrstuhl)

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

Blick über die Vorlesung heute

■ Organisatorisches

- Ihre Erfahrungen + Ergebnisse mit dem **Ü14 (Editierdistanz)**
- Ergebnis der offiziellen **Evaluation** dieser Vorlesung

■ Klausur

- Kurz zur Art der Aufgaben + Benotungsschema
- Ich werde ein paar Aufgaben live vorrechnen

■ Vorstellung Lehrstuhl

- Unsere Art zu arbeiten
- Aktuelle Projekte + was so alles dahinter steckt
- Weiterführende (Spezial-) Vorlesungen

Erfahrungen mit dem Ü14 (Editierdistanz)

■ Zusammenfassung / Auszüge Stand 12. Februar 16:00

- Interessantes / praxis-relevantes Thema
- Interessante Daten (AOL Query Log)
- Übungsblatt für die meisten gut und schnell machbar
- Ergebnistabelle wie immer gut und nützlich
- Viel über die Hintergründe alltäglicher Dinge gelernt

■ Teilnehmer / Rücklauf

- Aktive Teilnehmer/innen der Vorlesung: **33**
- Anmeldungen zur Klausur bisher: **27** ... nicht vergessen !
- An der Evaluation teilgenommen: **30** ... super !
- Davon **27** x ESE, **3** x IEMS
- Davon **4** x 1. Semester, **19** x 3. Semester, **7** x anderes
- Anzahl Lehrpreisnominierungen: **28** ... danke !

- Es folgt eine Zusammenfassung Ihrer Rückmeldungen
- Alle Details dazu auf dem Wiki:

<http://ad-wiki.informatik.uni-freiburg.de/teaching/AlgoDatEselemsWS1213/Feedback>

■ Vorlesungsstil

- Viel gelernt: 27 x stimme voll zu, 2 x stimme zu, 1 x teils
- Niveau der Vorlesung: 17 x hoch, 12 x angemessen, 1 x zu
- Gut erklärt: 24 x stimme voll zu, 5 x stimme zu, 1 x teils
- Geht auf Fragen ein: 25 x stimme voll zu, 5 x stimme zu
- Anwendungsbezogen / praxisnah / nicht so trocken
- Angenehme / lockere Atmosphäre, gutes Tempo
- Algorithmen auch implementiert
- Beste Vorlesung bisher / im 3. Semester

■ Übungsblätter

- Schwierigkeit: 17 x angemessen, 12 x schwierig, 1 x zu
- Sinnvolle Ergänzung: 25 x stimme voll zu, 4 x stimme zu
- Aufwand: 14 x 9-12 Stunden, 7 x weniger, 7 x mehr
- Sehr gut auf die Vorlesung abgestimmt
- Interessante Aufgaben / Praxisbezug ist gut

■ Materialien / Online Support

- Materialien hilfreich: 23 x stimme voll zu, 7 x stimme zu
- Forum sehr nützlich
- Schnelle Antwortzeiten, auch nachts und am WE
- Sehr gute Aufzeichnungen + schnell online
- VL-Konsum: 12 x Anwesenheit, 11 x Aufzeichnung, 7 x beides

■ Mecker

- Manchmal etwas langsam, ein Beispiel reicht
- Mehr Geschichtliches
- Extrastunden für Fragen zum Übungsblatt
- Einstieg **Ü1** schwierig / Vormachen in der ersten Vorlesung zu schnell ... danach aber alles gut
- Bessere Anleitung zu **SVN**, **Makefile**, **Linux**, ...
- Mehr Programmieren als alles andere
- Ressourcenhungrige Übungsblätter
- Wechsel zwischen **Java** und **C++** in VL ist nicht gut

■ Wann, wo, wie

- Samstag, 23. März 2013, 11:00 – 13:00 Uhr im HS 026
- 4 Aufgaben à jeweils 20 Punkte → K1, K2, K3, K4
- Summe Übungspunkte auf max. 20 Punkte skaliert → Ü
- Von Ü, K1, K2, K3, K4 werden die besten vier genommen und aufsummiert (also maximal 80 Punkte)
- Die Klausur ist **open book** : sie dürfen Bücher, Papier, usw. in beliebiger Menge mitbringen

Aber seien sie bitte sparsam beim Ausdrucken der Folien

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

■ Typen von Aufgaben

- **Typ 1:** Einen Algorithmus, oder eine Variante davon, an einem Beispiel nachvollziehen ... [siehe Buntstifte](#)
- **Typ 2:** Kleineres Programm schreiben, oder gegebenes Programm verstehen
- **Typ 3:** Kleinere Rechenaufgaben oder Beweise, insbesondere Induktionsbeweise ... [siehe Gehirn](#)

1.1

0	2	4	6	8
1	1	3	5	7
2				
3				

$$\vartheta(x) = x^2 \bmod 4$$

einsetz: 1, 2, 3, 4, 5, 6, 7, 8

$$\begin{matrix} 2 & \downarrow & 6 & \downarrow & 4 & \downarrow & 7 & \downarrow & 8 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & \end{matrix}$$

1.2

$$\times \text{ gerade} \Rightarrow x = 2 \cdot m, m \in \mathbb{N}$$

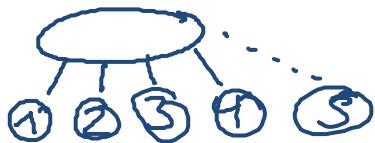
$$\Rightarrow \vartheta(x) = (2 \cdot m)^2 \bmod 4 = 4 \cdot m^2 \bmod 4 = 0$$

$$\times \text{ ungerade} \Rightarrow x = 2 \cdot m + 1, m \in \mathbb{N}$$

$$\Rightarrow \vartheta(x) = (2 \cdot m + 1)^2 \bmod 4$$

$$= \underbrace{4 \cdot m^2 + 4 \cdot m + 1}_{-4 \cdot (m^2 + m)} \bmod 4 = 1$$

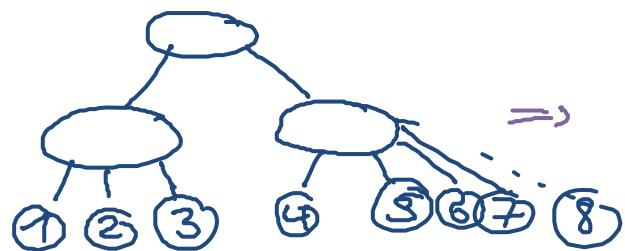
Aufgabe 2 (a, b) -Bäume



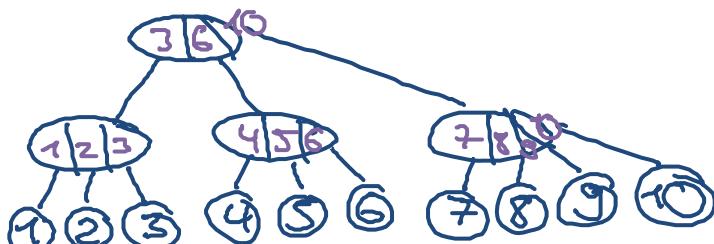
$(2, 4)$ -Baum

Insert: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

6



\Rightarrow



Aufgabe 4

O -Notation

$$\begin{aligned}
 & n, n \cdot \log n, 2 \cdot n \cdot \log n, \dots, n^2, n \cdot 2 \cdot \log n = n \cdot \log(n^2) = n \cdot \log(n^2) \\
 & \lim_{n \rightarrow \infty} \frac{x}{n \cdot \log n} = 0 \Rightarrow n = O(n \cdot \log n) \\
 & n \neq \Theta(n \cdot \log n)
 \end{aligned}$$

$$\frac{n \cdot \log n}{n^2}$$

$$n \cdot (\log n)^2$$

■ Unsere Arbeitsweise

- 1/3 Theorie (neue Algorithmen, Laufzeitanalyse, etc.)

Z.B. Algorithmen zur Berechnung von kürzesten Wegen in großen Straßennetzwerken ... Dijkstra ist da viel zu langsam

- 1/3 Algorithm Engineering (gute Implementierungen)

Viele Ideen klappen nur in der Theorie gut, erst beim Implementieren merkt man, ob es wirklich was bringt

Rumhacken ohne Verständnis der Theorie dahinter bringt aber auch nichts, da stochert man nur rum

- 1/3 Software Engineering (gute Software)

Ein Programm für sich alleine zu schreiben, das jetzt gerade einmal läuft ist eine Sache ... etwas mit anderen zusammen zu schreiben, dass auch in 5 Jahren noch läuft eine andere

■ Aktuelle Projekte

- Multi-modale Routenplanung
Beliebige Kombination von Auto, ÖPNV, Flugzeug, ...
Modelle dafür, effiziente Algorithmen, lauffähiges System
- Semantische Suche
Suche mit Verständnis der Sprache
Beispiel auf Google
Beispiele mit Broccoli: <http://broccoli.informatik.uni-freiburg.de>
- Automatische Analyse von Bewegungsdaten
Erkennen von Mustern, Zusammenhängen
Zuordnung zu Krankheitssymptomen / Neurodaten

■ Spezialvorlesungen

- **Information Retrieval** ... nächstes Mal im WS 13/14

Alles was man braucht um eine Suchmaschinen gemäß dem Stand der Kunst zu bauen

Potpourri aus vielen Techniken und Gebieten: Algorithmen, Kodierungstheorie, Web Apps, Masch. Lernen, Algebra, Statistik, ...

Wie immer: in Vorlesung erklärt, in Übung implementiert

- **Efficient Route Planning** ... nächstes Mal erst SS 2014

Alles was man braucht um einen Routenplaner gemäß dem Stand der Kunst zu bauen

Viele Algorithmen / Heuristiken und Ihre Implementierung