SS 2021

Algorithmen und Datenstrukturen

0. Kapitel

Vorbemerkungen

Martin Dietzfelbinger

April 2021

Team:

Folien und verantwortlich: Univ.-Prof. Dr. **Martin Dietzfelbinger**

Vorlesung & Übung & Praktikum:

M. Sc. **Philipp Schlag**

FG Komplexitätstheorie und Effiziente Algorithmen
`http://www.tu-ilmenau.de/ktea/`

Hörer/innen:

Informatikstudierende im 2. Semester.

Andere Interessierte *sehr* willkommen!

Hörer/innen:

Informatikstudierende im 2. Semester.

Andere Interessierte *sehr* willkommen!

Material:

Material und Kommunikation über

`https://moodle2.tu-ilmenau.de/course/view.php?id=3530`

Datenstrukturen!

Datenstrukturen!

- Stacks, Queues
- Arrays
- Lineare Listen

Datenstrukturen!

- Stacks, Queues
- Arrays
- Lineare Listen
- Suchbäume

Datenstrukturen!

- Stacks, Queues
- Arrays
- Lineare Listen
- Suchbäume (implementieren Wörterbücher)

Datenstrukturen!

- Stacks, Queues
- Arrays
- Lineare Listen
- Suchbäume (implementieren Wörterbücher)
- Balancierte Suchbäume

Datenstrukturen!

- Stacks, Queues
- Arrays
- Lineare Listen
- Suchbäume (implementieren Wörterbücher)
- Balancierte Suchbäume
- Hashtabellen (implementieren Wörterbücher)

Datenstrukturen!

- Stacks, Queues
- Arrays
- Lineare Listen
- Suchbäume (implementieren Wörterbücher)
- Balancierte Suchbäume
- Hashtabellen (implementieren Wörterbücher)
- Heaps (implementieren Prioritätswarteschlangen)

Datenstrukturen!

- Stacks, Queues
- Arrays
- Lineare Listen
- Suchbäume (implementieren Wörterbücher)
- Balancierte Suchbäume
- Hashtabellen (implementieren Wörterbücher)
- Heaps (implementieren Prioritätswarteschlangen)
- Union-Find-Datenstruktur

Datenstrukturen!

- Stacks, Queues
- Arrays
- Lineare Listen
- Suchbäume (implementieren Wörterbücher)
- Balancierte Suchbäume
- Hashtabellen (implementieren Wörterbücher)
- Heaps (implementieren Prioritätswarteschlangen)
- Union-Find-Datenstruktur
- Datenstrukturen für Graphen

Algorithmen!

Algorithmen!

- Sortieren: Mergesort, Quicksort, Heapsort

Algorithmen!

- Sortieren: Mergesort, Quicksort, Heapsort
- Graphdurchläufe: Breitensuche, einfache Tiefensuche

Algorithmen!

- Sortieren: Mergesort, Quicksort, Heapsort
- Graphdurchläufe: Breitensuche, einfache Tiefensuche
- Volle Tiefensuche, starke Zusammenhangskomponenten

Algorithmen!

- Sortieren: Mergesort, Quicksort, Heapsort
- Graphdurchläufe: Breitensuche, einfache Tiefensuche
- Volle Tiefensuche, starke Zusammenhangskomponenten
- Divide-and-Conquer: Multiplikation von ganzen Zahlen und von Matrizen, Auswahlproblem, Schnelle Fourier-Transformation (eventuell)

Algorithmen!

- Sortieren: Mergesort, Quicksort, Heapsort
- Graphdurchläufe: Breitensuche, einfache Tiefensuche
- Volle Tiefensuche, starke Zusammenhangskomponenten
- Divide-and-Conquer: Multiplikation von ganzen Zahlen und von Matrizen, Auswahlproblem, Schnelle Fourier-Transformation (eventuell)
- Greedy-Ansatz: Kürzeste Wege, Huffman-Codes, Minimale Spannbäume, fraktionales Rucksackproblem

Algorithmen!

- Sortieren: Mergesort, Quicksort, Heapsort
- Graphdurchläufe: Breitensuche, einfache Tiefensuche
- Volle Tiefensuche, starke Zusammenhangskomponenten
- Divide-and-Conquer: Multiplikation von ganzen Zahlen und von Matrizen, Auswahlproblem, Schnelle Fourier-Transformation (eventuell)
- Greedy-Ansatz: Kürzeste Wege, Huffman-Codes, Minimale Spannbäume, fraktionales Rucksackproblem
- Dynamische Programmierung: Kürzeste Wege, Editierdistanz, 0-1-Rucksackproblem

Algorithmen und Datenstrukturen: Ziele

Studierende sollen beherrschen:

- Grundlage für Analyse: Umgang mit **O-Notation**
- Eine **Spezifikation**stechnik für **Datentypen**
- Relevante **Implementierungen** fundamentaler **Datentypen**
- **Auswahlkriterien** für elementare **Datenstrukturen** kennen und anwenden
- Kenntnis grundlegender **Entwurfsstrategien** („Algorithmenparadigmen“)
- Kenntnis **grundlegender Algorithmen** und ihrer Eigenschaften
- **Auswahlkriterien** für **Algorithmen**, insbesondere Ressourcenverbrauch, kennen und anwenden
- Standard-**Analysemethoden** für Ressourcenverbrauch (Rechenzeit, Speicherplatz)
- **Korrektheitsbeweise** für Algorithmen verstehen und darstellen

Zur Arbeitsweise:

Stoff ist **viel** zu **kompliziert** und **umfangreich**, um durch reines Zuhören (bzw. einfaches Durchlesen) verstanden zu werden.

- Regelmäßig **Vorlesung nacharbeiten**. **Semesterbegleitend!**

Zur Arbeitsweise:

Stoff ist **viel** zu **kompliziert** und **umfangreich**, um durch reines Zuhören (bzw. einfaches Durchlesen) verstanden zu werden.

- Regelmäßig **Vorlesung nacharbeiten**. **Semesterbegleitend!**
- Begleitend Bücher ansehen.

Zur Arbeitsweise:

Stoff ist **viel** zu **kompliziert** und **umfangreich**, um durch reines Zuhören (bzw. einfaches Durchlesen) verstanden zu werden.

- Regelmäßig **Vorlesung nacharbeiten**. **Semesterbegleitend!**
- Begleitend Bücher ansehen.
- Übungsblätter drucken, lesen, zur Übung mitbringen,
  vorher Lösung ausdenken, Lösungsweg aufschreiben, an Lösungen **mitarbeiten**,
  Lösungen **vortragen**.

Zur Arbeitsweise:

Stoff ist **viel** zu **kompliziert** und **umfangreich**, um durch reines Zuhören (bzw. einfaches Durchlesen) verstanden zu werden.

- Regelmäßig **Vorlesung nacharbeiten**. **Semesterbegleitend!**
- Begleitend Bücher ansehen.
- Übungsblätter drucken, lesen, zur Übung mitbringen, **vorher Lösung ausdenken**, Lösungsweg aufschreiben, an Lösungen **mitarbeiten**, Lösungen **vortragen**.
- Regelmäßig **Übungen nacharbeiten**. **Semesterbegleitend!**

Zur Arbeitsweise:

Stoff ist **viel** zu **kompliziert** und **umfangreich**, um durch reines Zuhören (bzw. einfaches Durchlesen) verstanden zu werden.

- Regelmäßig **Vorlesung nacharbeiten**. **Semesterbegleitend!**
- Begleitend Bücher ansehen.
- Übungsblätter drucken, lesen, zur Übung mitbringen, **vorher Lösung ausdenken**, Lösungsweg aufschreiben, an Lösungen **mitarbeiten**, Lösungen **vortragen**.
- Regelmäßig **Übungen nacharbeiten**. **Semesterbegleitend!**
- Praktikumsaufgaben **vorbereiten**. **Semesterbegleitend!**

Zur Arbeitsweise:

Stoff ist **viel** zu **kompliziert** und **umfangreich**, um durch reines Zuhören (bzw. einfaches Durchlesen) verstanden zu werden.

- Regelmäßig **Vorlesung nacharbeiten**. **Semesterbegleitend!**
- Begleitend Bücher ansehen.
- Übungsblätter drucken, lesen, zur Übung mitbringen, **vorher Lösung ausdenken**, Lösungsweg aufschreiben, an Lösungen **mitarbeiten**, Lösungen **vortragen**.
- Regelmäßig **Übungen nacharbeiten**. **Semesterbegleitend!**
- Praktikumsaufgaben **vorbereiten**. **Semesterbegleitend!**
- Bei Verständnisproblemen **frühzeitig fragen!**

Zeitaufwand? – Beispielrechnung

Zeitaufwand? – Beispielrechnung

Leistungspunkte: 8 LP – Entspricht 240 Zeitstunden.

Zeitaufwand? – Beispielrechnung

Leistungspunkte: 8 LP – Entspricht 240 Zeitstunden.

Vorlesungszeit: 15 Wochen.

Zeitaufwand? – Beispielrechnung

Leistungspunkte: 8 LP – Entspricht 240 Zeitstunden.

Vorlesungszeit: 15 Wochen.

12 Stunden pro Woche

Zeitaufwand? – Beispielrechnung

Leistungspunkte: 8 LP – Entspricht 240 Zeitstunden.

Vorlesungszeit: 15 Wochen.

12 Stunden pro Woche

Davon: **6** in Vorlesung/Übung/Praktikum

Zeitaufwand? – Beispielrechnung

Leistungspunkte: 8 LP – Entspricht 240 Zeitstunden.

Vorlesungszeit: 15 Wochen.

12 Stunden pro Woche

Davon: **6** in Vorlesung/Übung/Praktikum

Zeitaufwand: **6–7** Zeitstunden pro Woche
neben Vorlesung/Übung!

Zeitaufwand? – Beispielrechnung

Leistungspunkte: 8 LP – Entspricht 240 Zeitstunden.

Vorlesungszeit: 15 Wochen.

12 Stunden pro Woche

Davon: **6** in Vorlesung/Übung/Praktikum

Zeitaufwand: **6–7** Zeitstunden pro Woche
neben Vorlesung/Übung!

ergibt: **180–195** Stunden.

Zeitaufwand? – Beispielrechnung

Leistungspunkte: 8 LP – Entspricht 240 Zeitstunden.

Vorlesungszeit: 15 Wochen.

12 Stunden pro Woche

Davon: **6** in Vorlesung/Übung/Praktikum

Zeitaufwand: **6–7** Zeitstunden pro Woche
neben Vorlesung/Übung!

ergibt: **180–195** Stunden.

Bleiben: **45–60 Stunden Prüfungsvorbereitung!**

Vorlesung/Übung

(Donnerstag 15:00 – 16:30 Audimax)
(Freitag 13:00 – 14:30 Audimax)

Vorlesung bis auf Weiteres ersetzt durch Online-Angebot.

(Dienstag 17:00 – 18:30 Sr HU 013)
(Mittwoch 9:00 – 10:30 Sr H 2509)
(Mittwoch 11:00 – 12:30 Sr H 1520a)

Übung vorläufig ersetzt durch Online-Angebot.

Übungsblätter stehen im Netz.
Übung **vor-** und **nach**bereiten!

Prüfungsstoff: Vorlesung + Übungsaufgaben.

Prüfungsstoff: Vorlesung + Übungsaufgaben.

Prüfungsklausur: 150 Minuten, **150 Punkte.**

Achtung: Angaben zu Klausuren
unter dem Vorbehalt der Durchführbarkeit im Juni bzw. Juli/August.

Praktikum

Verpflichtend nur für Informatikstudierende!

Praktikum

Verpflichtend nur für Informatikstudierende!

(Donnerstag, 7:00 – 8:30, gerade (G) Wochen, RTK 6)
(Donnerstag, 9:00 – 10:30, gerade (G) Wochen, RTK 6)

Praktikum

Verpflichtend nur für Informatikstudierende!

(Donnerstag, 7:00 – 8:30, gerade (G) Wochen, RTK 6)
(Donnerstag, 9:00 – 10:30, gerade (G) Wochen, RTK 6)
Präsenzangebot beginnt, sobald dies möglich ist.

Live-Programmierung in **C++:**
Implementierung und Zeitmessung (hauptsächlich) zu Algorithmen und Datenstrukturen der Vorlesung.
Praktikumsaufgaben/-anleitungen auf der Webseite.

Praktikum

Verpflichtend nur für Informatikstudierende!

(Donnerstag, 7:00 – 8:30, gerade (G) Wochen, RTK 6)
(Donnerstag, 9:00 – 10:30, gerade (G) Wochen, RTK 6)
Präsenzangebot beginnt, sobald dies möglich ist.

Live-Programmierung in **C++:**
Implementierung und Zeitmessung (hauptsächlich) zu
Algorithmen und Datenstrukturen der Vorlesung.
Praktikumsaufgaben/-anleitungen auf der Webseite.

Abschluss (Modulverpflichtung, ohne Note):
Präsentation ausgewählter Praktikumsaufgaben am Semesterende

Grundlagen, Voraussetzungen

Grundlagen, Voraussetzungen

- Prof. Schäfer, Algorithmen und Programmierung für (Ingenieur-)Informatiker, Vorlesung, WS 2019/20
  `https://www.tu-ilmenau.de/telematik/lehre/wintersemester-2019-2020/algorithmen-und-programmierung-fuer-ingenieur-informatiker/`
  (wird vorausgesetzt).

Grundlagen, Voraussetzungen

- Prof. Schäfer, Algorithmen und Programmierung für (Ingenieur-)Informatiker, Vorlesung, WS 2019/20
  `https://www.tu-ilmenau.de/telematik/lehre/wintersemester-2019-2020/algorithmen-und-programmierung-fuer-ingenieur-informatiker/`
  (wird vorausgesetzt).
- G. Saake, K. U. Sattler. Algorithmen und Datenstrukturen – Eine Einführung mit Java. dpunkt.verlag, 5. Auflage, 2013 (Referenz für Grundlagen, Hintergrund).

Grundlagen, Voraussetzungen

- Prof. Schäfer, Algorithmen und Programmierung für (Ingenieur-)Informatiker, Vorlesung, WS 2019/20
  `https://www.tu-ilmenau.de/telematik/lehre/wintersemester-2019-2020/algorithmen-und-programmierung-fuer-ingenieur-informatiker/`
  (wird vorausgesetzt).
- G. Saake, K. U. Sattler. Algorithmen und Datenstrukturen – Eine Einführung mit Java. dpunkt.verlag, 5. Auflage, 2013 (Referenz für Grundlagen, Hintergrund).
- Prof. Kriesell, Grundlagen und Diskrete Strukturen, Vorlesung, WS 2020/21
  (wird vorausgesetzt).

Grundlagen, Voraussetzungen

- Prof. Schäfer, Algorithmen und Programmierung für (Ingenieur-)Informatiker, Vorlesung, WS 2019/20
  `https://www.tu-ilmenau.de/telematik/lehre/wintersemester-2019-2020/algorithmen-und-programmierung-fuer-ingenieur-informatiker/`
  (wird vorausgesetzt).
- G. Saake, K. U. Sattler. Algorithmen und Datenstrukturen – Eine Einführung mit Java. dpunkt.verlag, 5. Auflage, 2013 (Referenz für Grundlagen, Hintergrund).
- Prof. Kriesell, Grundlagen und Diskrete Strukturen, Vorlesung, WS 2020/21
  (wird vorausgesetzt).
- Dr. Schreyer, Mathematik für Informatiker 1, Vorlesung, WS 2020/21
  (wird vorausgesetzt).

Literaturvorschläge:

- R. H. **Güting** und S. **Dieker**, **Datenstrukturen und Algorithmen**, 4. Auflage (erweitert), SpringerVieweg 2018.
  Einführend, kompakt, für Bachelorstudium. Themen überlappen sich mit Vorlesung, sind aber *nicht deckungsgleich*. Gut zum „Reinkommen“.
  eBook, Universitätsbibliothek, sogar Download. Im Katalog nach PPN 1028030827 suchen.
- T. H. **Cormen**, C. E. **Leiserson**, R. L. **Rivest**, C. **Stein**,
  Introduction to Algorithms, 3rd ed., MIT Press, 2009.
  Reicht für das ganze Studium und darüber hinaus.
  Deutsche Ausgabe, 4. Auflage, 2013 bei Oldenbourg.

Literaturvorschläge:

- R. H. **Güting** und S. **Dieker**, **Datenstrukturen und Algorithmen**, 4. Auflage (erweitert), SpringerVieweg 2018.
  Einführend, kompakt, für Bachelorstudium. Themen überlappen sich mit Vorlesung, sind aber *nicht deckungsgleich*. Gut zum „Reinkommen“.
  eBook, Universitätsbibliothek, sogar Download. Im Katalog nach PPN 1028030827 suchen.
- T. H. **Cormen**, C. E. **Leiserson**, R. L. **Rivest**, C. **Stein**,
  Introduction to Algorithms, 3rd ed., MIT Press, 2009.
  Reicht für das ganze Studium und darüber hinaus.
  Deutsche Ausgabe, 4. Auflage, 2013 bei Oldenbourg.
  Englisches eBook, Universitätsbibliothek: Im Katalog nach PPN 686807499 suchen.
- T. **Ottmann**, P. **Widmayer**, **Algorithmen und Datenstrukturen**,
  6. Aufl., SpringerVieweg, 2017.

Literaturvorschläge:

- R. H. **Güting** und S. **Dieker**, **Datenstrukturen und Algorithmen**, 4. Auflage (erweitert), SpringerVieweg 2018.
  Einführend, kompakt, für Bachelorstudium. Themen überlappen sich mit Vorlesung, sind aber *nicht deckungsgleich*. Gut zum „Reinkommen“.
  eBook, Universitätsbibliothek, sogar Download. Im Katalog nach PPN 1028030827 suchen.
- T. H. **Cormen**, C. E. **Leiserson**, R. L. **Rivest**, C. **Stein**,
  Introduction to Algorithms, 3rd ed., MIT Press, 2009.
  Reicht für das ganze Studium und darüber hinaus.
  Deutsche Ausgabe, 4. Auflage, 2013 bei Oldenbourg.
  Englisches eBook, Universitätsbibliothek: Im Katalog nach PPN 686807499 suchen.
- T. **Ottmann**, P. **Widmayer**, **Algorithmen und Datenstrukturen**,
  6. Aufl., SpringerVieweg, 2017. (Deutsches Standardwerk, sehr umfassend.)

Literaturvorschläge (Forts.):

- M. **Dietzfelbinger**, K. **Mehlhorn**, P. **Sanders**,
  Algorithmen und Datenstrukturen – Die Grundwerkzeuge, Springer-Verlag, 2014
  (Ist im Stil verschieden von der Vorlesung, Themen überschneiden sich, das Buch führt bei einigen Themen deutlich weiter; für Interessierte. Originalausgabe auf englisch von Mehlhorn und Sanders. Neues englisches Buch: Fortgeschrittener, hat auch parallele Algorithmen zum Thema.)

Literaturvorschläge (Forts.):

- M. **Dietzfelbinger**, K. **Mehlhorn**, P. **Sanders**,
  Algorithmen und Datenstrukturen – Die Grundwerkzeuge, Springer-Verlag, 2014
  (Ist im Stil verschieden von der Vorlesung, Themen überschneiden sich, das Buch führt bei einigen Themen deutlich weiter; für Interessierte. Originalausgabe auf englisch von Mehlhorn und Sanders. Neues englisches Buch: Fortgeschrittener, hat auch parallele Algorithmen zum Thema.)
- R. **Sedgewick** and K. **Wayne**, **Algorithms**, 4th ed., Addison-Wesley, 2012. (Amerikanisches Standardwerk, umfassend, programmiernah. Wie Cormen et al.: reicht für das ganze Studium. Ressourcen im Web: Video-Lectures, Aufgaben, Programmcode. Deutsch: 4. Auflage, Pearson, 2014, ebenfalls mit Web-Ressourcen, insbesondere Programmcode.)

Literaturvorschläge (Forts.):

- M. **Dietzfelbinger**, K. **Mehlhorn**, P. **Sanders**,
  Algorithmen und Datenstrukturen – Die Grundwerkzeuge, Springer-Verlag, 2014
  (Ist im Stil verschieden von der Vorlesung, Themen überschneiden sich, das Buch führt bei einigen Themen deutlich weiter; für Interessierte. Originalausgabe auf englisch von Mehlhorn und Sanders. Neues englisches Buch: Fortgeschrittener, hat auch parallele Algorithmen zum Thema.)
- R. **Sedgewick** and K. **Wayne**, **Algorithms**, 4th ed., Addison-Wesley, 2012. (Amerikanisches Standardwerk, umfassend, programmiernah. Wie Cormen et al.: reicht für das ganze Studium. Ressourcen im Web: Video-Lectures, Aufgaben, Programmcode. Deutsch: 4. Auflage, Pearson, 2014, ebenfalls mit Web-Ressourcen, insbesondere Programmcode.)
- J. **Kleinberg**, É. **Tardos**, **Algorithm Design**, Pearson Education, 2005 (weiterführend).

Literaturvorschläge (Forts.):

- M. **Dietzfelbinger**, K. **Mehlhorn**, P. **Sanders**,
  Algorithmen und Datenstrukturen – Die Grundwerkzeuge, Springer-Verlag, 2014
  (Ist im Stil verschieden von der Vorlesung, Themen überschneiden sich, das Buch führt bei einigen Themen deutlich weiter; für Interessierte. Originalausgabe auf englisch von Mehlhorn und Sanders. Neues englisches Buch: Fortgeschrittener, hat auch parallele Algorithmen zum Thema.)
- R. **Sedgewick** and K. **Wayne**, **Algorithms**, 4th ed., Addison-Wesley, 2012. (Amerikanisches Standardwerk, umfassend, programmiernah. Wie Cormen et al.: reicht für das ganze Studium. Ressourcen im Web: Video-Lectures, Aufgaben, Programmcode. Deutsch: 4. Auflage, Pearson, 2014, ebenfalls mit Web-Ressourcen, insbesondere Programmcode.)
- J. **Kleinberg**, É. **Tardos**, **Algorithm Design**, Pearson Education, 2005 (weiterführend).

Vorlesung folgt eigenem Plan, nicht direkt einem Buch.