

Algorithmen und Datenstrukturen

AD1: VU 186.813, 4h, 6 ECTS

AD2: VU 186.815, 2h, 3 ECTS



Grundstruktur

AD1 und AD2 bestehen jeweils aus

- ▶ Einem Vorlesungsteil
- ▶ Einem Übungsteil
- ▶ Einer Programmieraufgabe

Voraussetzungen

AD1 Vorlesungsteil und Übungsteil:

- ▶ Grundkenntnisse der linearen Algebra und Analysis (Mengenlehre, Metriken, Folgen und Reihen)
- ▶ Grundkenntnisse des Programmierens (Anweisungen, Schleifen, Verzweigungen, Rekursion, Funktionen, einfache Datentypen, Arrays, Listen)

AD2 Vorlesungsteil und Übungsteil:

- ▶ Inhalte der AD1 Vorlesung

AD1 Programmieraufgabe

- ▶ Inhalte der AD1 Vorlesung
- ▶ Grundlegende Programmierkenntnisse in Java

AD2 Programmieraufgabe

- ▶ Inhalte der AD1 und AD2 Vorlesung
- ▶ Grundlegende Programmierkenntnisse in Java

Grundsätzliches

Ziel von AD1 und AD2: Vermittlung von wichtigen Grundkenntnissen für die Entwicklung und die Analyse von Algorithmen und der zugrundeliegenden Datenstrukturen

- ▶ Hier verstehen wir Algorithmen als fundamentale Lösungsideen für Berechnungsprobleme. Ein Algorithmus kann verbal oder als Pseudocode angegeben werden.
- ▶ Die tatsächliche Implementierung von Algorithmen benötigt zusätzliche Konzepte des Software Engineerings (objektorientierte Konzepte, Programmiersprachen-spezifische Deklarationen, etc.). Diese sind weder Gegenstand noch Voraussetzung von AD.
- ▶ Nur für das Lösen der Programmieraufgaben ist Programmiersprachen-spezifisches Wissen erforderlich, aber die Art der Fragestellungen erfordert keine komplex strukturierten Programme.

AD1 Inhalte

- ▶ Einführung: Stable Matching
- ▶ Analyse von Algorithmen
 - ▶ Asymptotische Komplexität
 - ▶ Ω , O , Θ -Notationen
- ▶ Graphen und Graphenalgorithmen
 - ▶ Tiefen- und Breitensuche, topologisches Sortieren, Dijkstras Algorithmus, ...
- ▶ Algorithmen-Paradigmen
 - ▶ Greedy
 - ▶ Divide-and-Conquer
- ▶ Suche
 - ▶ Balancierte Suchbäume (AVL, B, B*)
 - ▶ Hashverfahren
- ▶ Datenstrukturen in der praktischen Anwendung

AD2 Inhalte

- ▶ Branch-and-Bound
- ▶ Approximation
- ▶ Dynamische Programmierung
- ▶ Heuristische Verfahren
- ▶ Polynomialzeitreduktion
- ▶ Effizient lösbare Spezialfälle schwerer Probleme

Organisation

Eingangstests

- ▶ AD1 und AD2 haben jeweils einen TUWEL-Eingangstest (multiple choice)

Vorlesung:

- ▶ AD1 und AD2 werden geblockt im Sommersemester abgehalten
- ▶ Vorlesungsteil mit zwei Einheiten pro Woche (Mi+Do)
- ▶ AD1: 16 Einheiten
- ▶ AD2: 10 Einheiten

Übungen

- ▶ AD1 hat 4 „Kreuzerl“-Übungseinheiten
- ▶ AD2 hat 2 „Kreuzerl“-Übungseinheiten

Übungstests

- ▶ AD1: drei schriftliche Tests, AD2: zwei schriftliche Tests

Programmieraufgaben

- ▶ AD1 und AD2: je eine Programmieraufgabe

Support

- ▶ Ausführliche Folien zum Download
- ▶ Video-Aufnahmen der Vorlesungen
- ▶ TUWEL-Forum
- ▶ Email-Hotline
- ▶ Sprechstunden (v.a. von Stefan Podlipnig)
- ▶ Repetitorien
- ▶ Fragestunden zu den Programmieraufgaben
- ▶ AD1: zwei TUWEL-Tests zur Teststvvorbereitung

- ▶ AD1: 606 Studierende (mit Zeugnis)
- ▶ AD2: 553 Studierende (mit Zeugnis)
- ▶ 25 parallele Übungsgruppen
- ▶ Arbeitsaufwand: > 2000 Stunden

AD neu ab SS 2016

- ▶ Material neu strukturiert, überarbeitet, und modernisiert
- ▶ Statt Skriptum ausführliche Folien (ca. 700 neugestaltete Folien)
- ▶ Besser an PK-Inhalte angepasst (z.B. verlinkte Listen und Binärsuche)
- ▶ Neue Übungsbeispiele

Status und Ausblick

- ▶ Neues Material scheint sehr gut zu funktionieren
- ▶ Wir haben schon positives feedback erhalten
- ▶ Es gibt konkrete Pläne zur Feinabstimmung für nächstes Jahr
- ▶ Wir sehen keinen Anlass für Änderungen in Inhalt oder Organisation