

Organisatorisches

Programmieren und Software-Engineering Theorie: Algorithmen, Homomorphismen und Formale Sprachen

22. Februar 2023

Organisatorisches

Die Gesamtnote POS ergibt sich durch

- POS Programmieren (nicht dieser Kurs)
- POS Theorie: (dieser Kurs!)
 - ① Wintersemester: “Graphentheorie”
 - ② Sommersemester:
 - **“Algorithmen”**
 - **“Homomorphismen”**
 - **“Formale Sprachen”**
 - **“Syntaxanalyse”**
- Alle Teile müssen positiv bestanden werden um positive Gesamtnote zu erhalten.

Beurteilung

- 65% SLÜs (Theorie)
- 35% Programmieraufgabe (Praxis)
- **Für eine positive Note müssen sowohl die Theorie als auch die Praxis (Programmieraufgabe) positiv sein.**
- Die SLÜs müssen gemeinsam eine positive Note ergeben, es muss jedoch nicht unbedingt jede einzelne SLÜ positiv sein.
- Bei nachweislicher Verhinderung bei der SLÜ kann auf Wunsch in der darauffolgenden Stunde eine Ersatzprüfung erfolgen.

Notenschlüssel

Prozent	Note
[0, 50]	Nicht Genügend
(50, 62.5]	Genügend
(62.5, 75]	Befriedigend
(75, 87.5]	Gut
(87.5, 100]	Sehr Gut

Programmieraufgabe

- Erstellen Sie ein Programm in Java, C#, Python, Javascript (andere Programmiersprachen nach Rücksprache möglich)
- Das Programm soll die Adjazenzmatrix eines Graphen aus einer Datei (csv) einlesen
- Weiters soll das Programm folgende Berechnungen durchführen (Minimalanforderungen):
 - Bestimmung der Distanzen und Exzentrizitäten aller Knoten
 - Radius, Durchmesser, Zentrum
 - Komponenten, Artikulationen, Brücken
- Wählen Sie eine dieser (oder andere) Erweiterungen für ein “Sehr Gut”:
 - Grafische Benutzeroberfläche
 - Eulersche Linien/Zyklen
 - Spannbäume/Gerüste
 - Starke Zusammenhangskomponente
 - Blöcke (schwierig!)
 - Test auf Isomorphie

Programmabgabe

- Fertigstellung des Programmes bis 1. Juni für Wertung im laufenden Semester
- Für die Inskription des folgenden Sommersemesters (z.B. 6AKIF) Terminvereinbarung bis 10. Jänner!
- Persönliches Abgabegespräch (Raum B4.18a), Dauer ca. 20 Minuten
- Terminvereinbarung per Mail, bzw. wird es Anfang Juni wird es eine Liste mit Zeit-Slots geben
- Quellcode als ZIP-Datei (keine Binärdateien)
- Namenskonvention (Beispiel): 2021-06-21__3BAIF_Gruber.zip
- Nur sinnvoll viele Kommentare im Code (keine vollständigen Erklärungen zu jeder Code-Zeile)!

Abgabegespräch

- **Raum B4.18a**
- Bereiten Sie mindestens einen Testfall mit mindestens 10 Knoten und mindestens zwei Brücken vor. Eine Zeichnung des/der Graphen soll ebenso präsentiert werden.
- Führen Sie zunächst das Programm aus: Demonstration der Funktionalität und Korrektheit.
- Geben Sie einen kurzen Überblick über den Aufbau des Programmes.
- Erklärung und Diskussion ausgewählter Algorithmen.
- Wie große Graphen kann ihr Programm berechnen? (Minimalanforderung: 15 Knoten)
- Feedback.

Kolloquium, spätere Programmabgabe

- Für reine Kolloquien (Theorie) nutzen Sie bitte einen der Sammeltermine (üblicherweise nach Semesterstart, in der Mitte und am Ende des Semesters)
- Terminvereinbarung per Mail (Betreff: Kolloquium 4XYIF POS)
- Bitte Klasse und Schuljahr im Mail anführen!
- Bei Programmabgaben oder für individuelle Termine zu Kolloquien: bitte zwei bis drei Termine (kompatibel mit unseren Stundenplänen) vorschlagen
- Zeit und Ort von Kolloquien (Sammeltermine) sind kurzfristig in meinem(!) Stundenplan ersichtlich. Im Zweifelsfall (falls nicht eingetragen) Treffpunkt in B4.18 ca. 10 Minuten vor Beginn.
- Bitte ausschließlich die Schul-Emailadresse verwenden, und bei Antworten immer die vorangegangene Kommunikation zitieren.
- Emailanfragen ohne entsprechende Informationen, bzw. zu hier beantworteten Fragen werden nicht beantwortet.

Stoffübersicht

- Algorithmen
 - Analyse
 - Rekursion
 - Datenstrukturen
 - ...
- Homomorphismen
- Syntaxanalyse
 - Grammatiken
 - Backus-Naur Form
 - Syntaxdiagramme, Syntaxanalyse
 - ...
- Formale Sprachen
 - Chomsky Hierarchie
 - Reguläre Sprachen, endliche Automaten
 - Kontextfreie Sprachen
 - Komplexität und Berechenbarkeit

Unterlagen

- Die Vortragsfolien sowie ein zur Verfügung gestelltes Programm decken die Inhalte ab!
- Eigene Mitschrift zu Beispielen, Erklärungen
- Einige Übungsbeispiele werden nur im Unterricht an der Tafel präsentiert.

Achtung!

Die Präsentationsfolien enthalten Animationen die zu gewissen Beispielen/Algorithmen eine Schritt-für-Schritt Erklärung enthalten. Diese Animationen sind am Besten im Vollbildmodus anzusehen! Im gedruckten Handout finden sich lediglich verkürzte Darstellungen mit weniger Zwischenschritten.

Weiterführende Inhalte (*)

- Manche Inhalte sind speziell für besonders Interessierte gedacht
- Diese Inhalte zählen nicht zum Kernstoffgebiet, und müssen somit nicht gelernt/gekonnt/verstanden werden
- Die Kennzeichnung dieser Inhalte erfolgt durch die blaue Titel- und Fußzeile in den Folien!

Literaturübersicht I

- [1] **Berger, Krieger, Mahr: “Grundlagen der elektronischen Datenverarbeitung”, Skriptum**
- [2] Dirk W. Hoffmann: “Theoretische Informatik”, Hanser, 3. Auflage
- [3] Gernot Salzer: “Einführung in die Theorie der Informatik”, Skriptum, TU Wien, 2001
- [4] Wikipedia (Englisch): <https://en.wikipedia.org/>
- [5] Wikipedia (Deutsch): <https://de.wikipedia.org/>
- [6] HappyCoders (Deutsch):
<https://www.happycoders.eu/de/algorithmen/>