

---

## 392041 Implementieren von Algorithmen (Ü) (WiSe 2020/2021)

---

Termin: Mi 08-10

Raum: GZI & ONLINE

Zeitraum: 26.10.2020-12.02.2021

### Motivation

Algorithmen sind das Handwerkszeug eines Informatikers, wenn es darum geht, viele theoretische, aber auch real vorkommende Probleme mithilfe des Computers zu lösen. Bereits in den ersten Semestern des Informatikstudiums lernen die Studierenden viele nützliche Datenstrukturen und Algorithmen kennen, die sie auf typische Probleme der Informatik anwenden können. Dies umfasst unter anderem klassische Datenstrukturen wie Listen, Bäume und Graphen sowie Algorithmen zur Suche nach kürzesten Pfaden, zum Textvergleich oder zur Kompression. Während die meisten Veranstaltungen sich auf das theoretische Verständnis dieser Algorithmen konzentrieren, liegt der Fokus in diesem Kurs auf der tatsächlichen Implementierung der Algorithmen in einer gängigen Programmiersprache, wie es in Projektarbeiten und der Berufspraxis erforderlich ist.

### Kursbeschreibung

Im Rahmen dieses Kurses haben Studierende die Möglichkeit, einen Algorithmus ihrer Wahl in einer ihnen bekannten Programmiersprache zu implementieren. Dies kann ein Algorithmus sein, den sie bereits im Verlauf ihres Studiums kennengelernt, aber noch nicht selbst implementiert haben, oder aber auch ein ganz neuer Algorithmus aus der aktuellen Forschung. Die Themen richten sich nach den Wünschen der Studierenden und werden in der ersten Sitzung individuell verteilt. In den darauffolgenden Sitzungen stellt immer eine/r der Studierenden zunächst kurz den Algorithmus und anschließend seine Implementierung vor und erhält Feedback von den anderen Teilnehmenden. Das Hauptaugenmerk soll sich dabei auf die Effizienz der Implementierung richten, d.h. auf die Frage, ob die theoretisch erzielbaren kleinsten oberen Schranken für Laufzeit und Speicher durch die Implementierung eingehalten werden. Die Studierenden sollen ein Gefühl für die effiziente Programmierung bekommen und auf typische Problemquellen wie verschachtelte Schleifen und Rekursionen im eigenen Code aufmerksam werden.

### Lernziele

Nach der erfolgreichen Teilnahme an diesem Kurs können die Studierenden...

- einige grundlegende Algorithmen der (Bio)Informatik *benennen* und *beschreiben*.
- die Funktionsweise dieser Algorithmen anhand von einfachen Beispielen *erklären*.
- einen Algorithmus effizient in der Programmiersprache ihrer Wahl *implementieren*.
- über geeignete Datenstrukturen und Strategien zur Implementierung *entscheiden*.
- das Laufzeit- und Speicherverhalten im eigenen und fremden Code *untersuchen*.
- ihren eigenen Code *präsentieren* sowie anderen konstruktives *Feedback geben*.

## Modulzuordnungen

- Modul 39-Inf-AB Algorithmen der Bioinformatik
- Modul 39-Inf-SAB Spezielle Algorithmen der Bioinformatik  
(oder *individueller Ergänzungsbereich*: 1 LP)

## Empfohlene Vorkenntnisse

- Modul 39-Inf-1 Algorithmen und Datenstrukturen
- Modul 39-Inf-2 Objektorientierte Programmierung

## Studien- und Prüfungsleistungen

- *Programmierung mit Präsentation (Studienleistung)*

Die Studierenden wählen einen Algorithmus aus, den sie im Laufe des Semesters in einer Programmiersprache ihrer Wahl implementieren. Die Bearbeitung erfolgt weitestgehend selbstständig und sollte einen Umfang von etwa 30 Stunden haben.

Die anschließende Präsentation sollte folgendermaßen aufgebaut sein:

- *Eine kurze Vorstellung des Algorithmus (3-5 Folien)*

1. Wer hat ihn erfunden? Welches Problem sollte damit gelöst werden?
2. Wie funktioniert der Algorithmus? Beschreibung und/oder Pseudocode.
3. Wie ist die theoretische Laufzeit- und Speicherkomplexität? Wieso?
4. Ein kleines Beispiel, um den Algorithmus „per Hand“ auszuprobieren.

- *Eine Vorstellung der Implementierung (Quellcode)*

Bei der Bewertung wird vor allem auf die Effizienz der Implementierung geachtet. Die Studierenden sollten ihren Code mindestens eine Woche vor der Präsentation zuschicken, um Feedback und eine Möglichkeit zur Nachbesserung zu erhalten.

- Eine *aktive Teilnahme* an der Veranstaltung ist wünschenswert. Die Studierenden sollen sich dazu gegenseitig Feedback geben.

## Themenauswahl

- Boyer-Moore-Algorithmus
- Knuth-Morris-Pratt-Algorithmus
- Gotoh-Algorithmus
- Hirschberg-Algorithmus
- Sellers-Algorithmus
- Center-Star-Algorithmus
- Waterman-Algorithmus
- Manber-Myers-Algorithmus
- WOTD-Algorithmus
- Neighbour Joining
- Agglomerative Clustering
- Fitch-/Sankoff-Algorithmus
- Kruskal-/Prim-Algorithmus
- Max-Flow-/Min-Cut-Theorem
- Burrows-Wheeler-Transformation
- Nussinov-Algorithmus
- Double-Cut-and-Join-Distanz
- De-Bruijn-Graphen

## Literatur

- T. Cormen, C. Leiserson: Algorithmen · Eine Einführung [↗](#)
- M. Nebel, S. Wild: Entwurf und Analyse von Algorithmen [↗](#)
- D. Gusfield: Algorithms on Strings, Trees and Sequences [↗](#)