

1. Semester

Grundlagen der funktionalen Programmierung, Datentypen und Funktionen in Haskell, einfache rekursive Funktionen
rekursive Listendefinitionen, Musteranpassung,
Spezifikation und Implementierung von Listenanwendungen, einfache Sortieralgorithmen
Suchen und Sortieren, effektive Algorithmen, Abschätzung des Aufwandes
Bäume und Anwendungen
Dynamische Datenstrukturen: verkettete Listen, Anwendungen
Bäume, Suchbäume, ausgeglichene Bäume: AVL-Bäume
Anwendungen: Huffman-Kodierung und andere
Hashtabellen als Datenstruktur
Graphen, Grundlagen, Arten, Repräsentationen, Algorithmen
Algorithmen auf Graphen: Suchalgorithmen: TS, BS mit den jeweiligen Datenstrukturen, Suche nach minimalen Wegen,
minimalen Spannäumen, Zyklensuche, (Algorithmen von Dijkstra, Kruskal, Prim)

2. Semester

Programmierprojekt: Themenfindung, Qualitätskriterien für Software
Programmierprojekt: Software-Life-Cycle, Implementierung
Programmierprojekt: Implementierung und Tests
Grafik: Pixelgrafik, Vektorgrafik, Repräsentation, Anwendungen, Vorteile, Nachteile
Algorithmen auf Pixelgrafiken: Drehen, Spiegeln, Invertieren, Filtern
Algorithmen zur Rasterung von Grafiken (Bresenham)
Kryptologie, Sicherheit und Datenschutz
symmetrische und asymmetrische Verfahren
RSA mit Grundlagen
Zero-Knowledge-Protokoll
Block-Chiffre (simple-DES)
Anforderungen an eine sichere Nachrichtenübertragung

nur noch 2. Semester

Netzwerke: Internet, Aufbau, Geschichte, grundlegende Prinzipien (dezentraler Aufbau, paketorientierte Verbindungen)
Schichten und Protokolle: OSI-Modell, Prinzip und Bedeutung, RFC
Grundlagen des Datenverkehrs auf den Schichten 3 (IP) und 4 (TCP, UDP)
Analyse und Implementierung von Beispielanwendungen
Sicherheit im Netz,

3. Semester

Datenbanken
Automaten und Sprachen

4. Semester

Haskell und funktionale Programmierung,
Softwareprojekt
Methoden zum Lösen kombinatorischer Probleme: Backtracking, gierige, genetische Algorithmen