

1. *Digitale Grundlagen*: Zahlensysteme (Dezimal, Binär, Hexadezimal); Darstellung von Zahlen (Integer); Zeichenketten (ASCII/Unicode); Bitmap- und Vektor-Grafiken
2. *Algorithmusbegriff*: Definition (Umschreibung); Laufzeitkomplexität (Beispiele, Aufwandabschätzungen); Darstellung von Algorithmen (natürliche Sprache, Flussdiagramm, Pseudocode, Programmcode); Beispiele: Euklidischer Algorithmus, Minimum Spanning Tree (Algorithmen von Prim und von Kruskal), Travelling Salesman Problem
3. *Datenstrukturen*: Array (Matrizen), Stack, Queue, Linked Lists, Bäume, Graphen (Standardoperationen, Anwendungen, Elemente der Implementierung)
4. *Programmieren mit Python*: Elementare Datentypen (`int`, `float`, `bool`), zusammengesetzte Datentypen (`list`, `tuples`, `dict`), Zuweisungen, Verzweigungen, Schleifen, Rekursion, Module, OOP (mit Cheat Sheet)
5. *Sortieralgorithmen*: Gnomesort, Selectionsort, Insertionsort, Bubblesort, Quicksort; manuelles Sortieren kleiner Listen, Sortieralgorithmen am Python-Code erkennen, Laufzeitkomplexität (Best und Worst Case)
6. *Datenbanken*: Relationales Datenmodell, Entity-Relationship-Model (ERM), Abbildungsregeln (als „Ersatz“ für Normalformen), relationale Algebra (Vereinigungsverträglichkeit, Vereinigung, Durchschnitt, Differenz, kartesisches Produkt, Projektion, Selektion, Verbund), DB-Abfragen mit SQLite (Cheat Sheet für SQL-Select-Pipeline)
7. *Deterministische endliche Automaten (DFA)*: Alphabete, Wörter und Sprachen; Akzeptieren von Sprachen durch DFAs; Konstruktion von DFAs; Anwendungen: Überprüfen von Zeichenklassen (z.B. ganze Zahlen) und Textsuche
8. *Suchalgorithmen*: lineare Suche, binäre Suche, Suche in Bäumen, Suche in Graphen (Tiefen- und Breitensuche), Stringmatching (naiv, Boyer-Moore-Horspool, DFA)
9. *Datenkompression*: Arten von Datenkompression, Lauflängencodierung, Huffman-Codierung, LZW-Codierung
10. *Simulationen*: Nagel-Schreckenbergs-Modell
11. *Kryptologie*: Hauptaufgaben; Kryptographie und Kryptoanalyse; Prinzip von Kerckhoffs; Substitutions- und Transpositionschiffren, symmetrische- und asymmetrische Kryptosysteme; Verfahren: Cäsar, Vigenère, One-Time-Pad; Multiplikation in \mathbb{Z}_n^* , eulersche φ -Funktion, erzeugende Elemente, Satz von Euler-Fermat, modulares Potenzieren mit Square and Multiply; Diffie-Hellman-Schlüsselaustausch; Schwachstellen der oben genannten Verfahren.
12. *Netzwerke*: Klassifikation nach Übertragungstechnik und Ausdehnung; Schichtenmodelle (Motivation, Protokolle, Dienste und Schnittstellen); 5-Schichten-Modell nach Tanenbaum: 1. *Bitübertragungsschicht (Bits)*: Funktion, Übertragungstechnologien, NRZ- und Manchesterkodierung; 2. *Sicherungsschicht (Rahmen)*: Funktion, Fehlererkennung und Fehlerkorrektur (Paritätsbit, Repetitions-codes); 3. *Vermittlungsschicht (Pakete)*: Funktion, IP-Adressen, Verbindungsarten, Routing, Algorithmus von Dijkstra; 4. *Transportschicht (Segmente)*: Funktion, Transportadressen; 5. *Anwendungsschicht (Nachrichten)*: Funktion, DNS, E-Mail, HTTP(S), ...