

Theoretische Grundlagen der Informatik, Algorithmen und Datenstrukturen

Vorläufige Themenübersicht SoSe 2025

June 10, 2025

Kurze Information zur Notation:

Hinter den Stichpunkten ist eine runde Klammer, die folgende Form und Bedeutung hat:

- (x), In Kapitel/Foliensatz x ist das Thema zu finden
- (x-y), In Kapitel/Foliensatz x ist das Thema auf/**ab** Folie y zu finden

1 Automatentheorie

Wissen und Verständnis:

- Alphabet (5)
- Wörter, Wortlänge (5)
- Präfix, Suffix, Infix (5)
- kanonische Ordnung (5)
- Sprachen (5)
- Graphen (6):
 - Gerichtet
 - Ungerichtet
 - Gewichtet
 - Nachbarschaft: Adjazent, Inzident
 - Schleifenfrei
 - Vollständig
 - Pfad

- Kreis
- Eulerischer Kreis
- Hamiltonscher Kreis
- Adjazenzmatrix
- Adjazenzliste
- Entscheidungsproblem (8)
- Wortproblem (8)
- Sprache ist rekursiv \equiv Sprache entscheidbar $\equiv \exists$ Aufzählalgorithmus (8)
- Optimierungsproblem (8)

Fähigkeiten:

- Induktive Definitionen (5)
- Codierung von Daten: Zahlen, Folgen, Matrizen (6)
- Abzählende Kombinatorik (7)

1.1 Deterministische endliche Automaten (DEAs)

Wissen und Verständnis:

- Definition (9)
- Konfiguration (9)
- Schrittrelation (9)
- Berechnung eines DEA (9)
- Kongruenzklasse (9)
- Äquivalente Zustände (10)
- Äquivalente Automaten (10)

Fähigkeiten:

- Automat erstellen (9)
- Automat benutzen (9)
- Minimierung von DEAs (10-23)

1.2 Nichtdeterministische endliche Automaten (NEAs)

Wissen und Verständnis:

- Definition (11)
- Konfiguration (11)
- Schrittrelation (11)
- Berechnung eines NEA (11)
- ϵ -Transitionen und ϵ -NEA (11)
- Grenzen regulärer Sprachen, was geht nicht mit DEA oder NEA? (11)
- Speicherbedarf vs. Laufzeit (12)
- Reguläre Ausdrücke und der Zusammenhang mit Automaten (13)

Fähigkeiten:

- Verknüpfung von Automaten (11)
- L^R mit ϵ -NEA erkennen (11-18)
- NEA zu DEA: Potenzmengenkonstruktion (12)

2 Laufzeit von Algorithmen

Wissen und Verständnis:

- Definition von \mathcal{O} , Ω , Θ (14.5)
- Best-case, Worst-case, Average-case (14.5)
- obere Schranke, untere Schranke (14.5)

Fähigkeit

- Definition von \mathcal{O} , Ω , Θ (14.5)
- Funktionen asymptotisch miteinander vergleichen (14.5)

2.1 Rekursionsgleichungen lösen

Fähigkeit:

- Master-Theorem (15-21)
- Erzeugende Funktionen (15)

3 Algorithmen&Datenstrukturen Grundlagen

Wissen und Verständnis:

- ADT (16-11,15)
- Sequenzen mit erforderlichen **Funktionen** und **Laufzeit**:
 - statische: Arrays/Felder (16-24)
 - dynamische Sequenzen: Verkettete Listen und dynamische Arrays/Felder (16-28)
 - Vergleich der Laufzeiten (16-53 und 56)
- Stack, LIFO (16-59)
- Queue, FIFO (16-63)
- Bäume und dazugehörige Begriffe (16-71)
- Baumdurchläufe (inklusive level-order) (16-89)

Fähigkeiten:

- Sequenzen bedienen:
 - Umgang mit verketteten Listen (16-46)
 - Datenstrukturen mit anderen Datenstrukturen simulieren (16)
 - Baumparameter berechnen (16-78)
 - Baumdurchläufe (inklusive level-order) (16-89)

3.1 Entwurfsmethoden

Wissen und Verständnis:

- Divide and Conquer/Teile und Herrsche mit den drei Schritten (17-5)
- Backtracking (17-10)
- Dynamische Programmierung und die bottom-up Berechnung (17-24)
- Greedy (17-41)

Fähigkeiten:

- Erkennen welcher Ansatz in einem Algorithmus benutzt wird (17)
- Algorithmus mit gegebenem Problem und Ansatz entwickeln (17)
- Entscheiden, ob ein Ansatz unpassend ist; Gegenbeispiele finden (17)

4 Sortierverfahren

Wissen und Verständnis:

- Klassifikationskriterien für Sortierverfahren (19Pflug-6)
- Laufzeiten und Eigenschaften der Sortierverfahren (19Pflug)
- Definition eines Heaps (19Pflug-67)

Fähigkeiten:

- Insertion-Sort anwenden/reproduzieren (19Pflug-26)
- Quick-Sort anwenden/reproduzieren (19Pflug-35)
- Heap verwenden (19Pflug-67)
- Heap-Sort anwenden/reproduzieren (19Pflug-79)
- Merge-Sort anwenden/reproduzieren (19Pflug-79)

4.1 Nicht Vergleichsbasierte Sortierverfahren

Wissen und Verständnis:

- Was bedeutet vergleichsbasiert (19Striegnitz-178)
- Ergebnis und Konsequenzen der unteren Schranke (19Striegnitz-178)
- Prinzipien der drei Sortierverfahren (19Striegnitz-182)

Fähigkeiten:

- Counting-Sort anwenden/reproduzieren (19Striegnitz-182)
- Bucket-Sort anwenden/reproduzieren (19Striegnitz-189)
- Radix-Sort anwenden/reproduzieren (19Striegnitz-197)

5 Suchen

Wissen und Verständnis:

- Binäre Suche Prinzip und Laufzeit (20-7)
- Hashing Prinzip und Anforderung (20-22)
- Problem Kollision und Lösungsstrategien (20-39)
- Belegungsfaktor (20-41)

- Offenes Hashing vs. geschlossenes Hashing (20-39)
- lineares, quadratisches Sondieren und Doppelhashing (20-45)

Fähigkeiten:

- Binäre Suche anwenden/reproduzieren (20-17)
- Eine gegebene Hashfunktion benutzen (20)
- Eine gegebene Sondierungsmethode anwenden (20-42)

5.1 Suchbäume

Wissen und Verständnis:

- Binäre Suchbäume (20-56)
- Laufzeiten für Operationen, abhängig von Baumparametern (20-56)
- Warum sind balancierte Suchbäume wünschenswert? (20-73)
- Definition AVL-Bäume (20-82)
- Definition von B-Bäumen

Fähigkeiten:

- Binäre Suchbäume einfügen/löschen (20-61)
- Leveldifferenz bei AVL-Bäumen bestimmen (20-88)
- AVL-Baum: einfügen und löschen
- B-Baum: einfügen und löschen

5.2 Mustersuche in Text

Wissen und Verständnis:

- String-Matching Problem (21)

Fähigkeiten:

- Größter Rand $\pi[i]$ für ein Muster bestimmen (21)
- Verfahren von Knuth, Morris und Pratt (KMP) anwenden (21)
- Verfahren von Boyer und Moore (BM) mit einfacher Bad Character-Heuristik anwenden (21.5)

6 Algorithmen auf Graphen

Wissen und Verständnis:

- Definitionen und Begriffe von Graphen (22)
- Definitionen und Begriffe von Fluss-Netzwerken (22)
- Matching Problem (22-191)
- minimaler Spannbaum (MST) Problem (22-198)
- Hamiltonscher Kreis Problem (22-237)
- Metrisches Traveling Salesman Problem (22-266)

Fähigkeiten:

- Breitensuche (BFS), Tiefensuche (DFS) anwenden (22)
- DAG topologisch sortieren (22)
- Algorithmus von Dijkstra (Single-Source Best Path) anwenden (22-74)
- Algorithmus von Bellman und Ford (Single-Source Best Path) anwenden (22-97)
- Algorithmus von Floyd-Warshall (All-Pairs Best Path) anwenden (22-135)
- Algorithmus von Ford-Fulkerson (Fluss) anwenden (22-181)
- Reduktion von Fluss auf Matching (Fluss) (22-194)
- Algorithmus von Kruskal (MST) anwenden (22-210)
- Algorithmus von Prim (MST) anwenden (22-220)
- 2-Approximation von TSP anwenden (22-267)
- 1.5-Approximation von Christofides (22-283)