

Zusammenfassung: Algorithmik

Kapitel 1: Einführung in die formale Behandlung von Algorithmen

Vergleich von grundlegenden Sortieralgorithmen (Funktionsweise und Laufzeit)

Landau-Symbole: Definition, Hierarchie, Vergleiche unterschiedlicher Maße, Bedeutung für die Komplexität von Algorithmen und Problemen, obere und untere Schranken.

Aufstellung von rekursiven Gleichungen für einfache Algorithmen (z.B. Mergesort, Quicksort) und Beweis des asymptotischen Laufzeitverhaltens aus diesen Gleichungen mit vollständiger Induktion.

~~Master-Theorem für Laufzeitabschätzungen aus rekursiven Gleichungen.~~

RAM: grundlegendes Verständnis, Bedeutung für Effizienzbewertung.

Beweis der unteren Schranke für das Sortierproblem mittels Vergleichen.

Kapitel 2: Weitere Such- und Sortieralgorithmen

Auswahlproblem: Randomisierter und deterministischer Algorithmus,

Rekursionsgleichungen für die Laufzeit, Kenntnis und Beweis (nur Idee) der Komplexität.

Suchen in sortierten Feldern: Binärsuche, Interpolationssuche, quadratische Binärsuche:

Funktionsweise, ~~Laufzeitbeweise~~.

Sortieren in endlichen Universen:

Countingsort, Radixsort, Bucketsort: Funktionsweise, Kenntnis der Komplexität

Zusammenfassung: Algorithmik

Kapitel 3: Lösungen für das Wörterbuchproblem

Definition des Wörterbuchproblems, Laufzeitziele.

~~Hashing: Grundidee, Vergleich mit anderen Verfahren.~~

2-3-Bäume: genaue Funktionsweise, Verallgemeinerung zu a-b-Bäumen.

~~AVL-Bäume~~, Rot-Schwarz-Bäume: Transformation zu 2-4-Bäumen.

~~Trie-Bäume, B-Bäume.~~

Algorithmus von Bellmann: Funktionsweise, abstrakt und am Beispiel,
Laufzeitabschätzung mit Beweisskizze,

~~Verbesserung von Knuth~~

Zusammenfassung: Algorithmik

Kapitel 4: Graphenalgorithmen

Grundstrukturen Heap und Union-Find mit Laufzeiten und Anwendungen in der Implementierung von Graphenalgorithmien

Genaue Funktionsweise, Laufzeitabschätzung ~~und Korrektheitsbeweis~~ des Algorithmus von Kruskal.

Genaue Funktionsweise, Laufzeitabschätzung ~~und Korrektheitsbeweis~~ des Algorithmus von Dijkstra.

All Pair Shortest Path-Problem: Algorithmus von Floyd-Warshall (mit Laufzeit und Korrektheitsbeweis)

Maximale Flüsse:

Begriffswelt: Erweiterungsweg, Restegraph, Levelgraph.

Satz von Ford-Fulkerson (Min-Schnitt/Max-Fluss): nur triviale Beweisrichtungen

Algorithmus von Edmonds-Karp: genaue Funktionsweise, Beweis: nur Überblick klausurrelevant

Algorithmus von Dinic: genaue Funktionsweise, Beweis: nur Überblick klausurrelevant

Erklärung der beiden Flussalgorithmen am Beispiel

~~Verschiedene Varianten der Definition des Matchingproblems für k (kDM), Komplexitätsklassen.~~
Bipartites Matching: Zusammenhang zu Flussproblem (Transformation).

Zusammenfassung: Algorithmik

Kapitel 5: String Matching

Naives Vorgehen: Laufzeit.

Algorithmus von Knuth-Morris-Pratt: genaue Funktionsweise (inkl. Präfixfunktion) am Beispiel,
Laufzeitabschätzung: wesentliche Idee (Akkumulatorfunktion)

Kapitel 6: Grundlagen der algorithmischen Geometrie

Basisprobleme Closest reference point, closest pair, minimum spanning tree, convex hull:

Triviale Algorithmen mit Laufzeitverhalten.

Voronoi-Diagramm: Definition und Bestandteile, Anwendungen auf Basisprobleme mit Laufzeit.

Zusammenhang zu Delaunay-Triangulierung.

Verfahren von Kirkpatrick: nur die grobe Idee, Rekursionsgleichung und Laufzeiten

Sweep-Verfahren: Grundidee, charakteristische Eigenschaften,

Erklärung der Begriffe an folgenden Anwendungen:

Closest Pair für $d=1,2$.

Bestimmung des Voronoi-Diagramms

Genauer Kenntnis der Invarianten des Plane Sweep für Voronoi-Diagramme, Laufzeitverhalten,
keine Beweise, keine Details der Einfügeoperation in die SSS