

Proseminar Algorithms and Data Structures Einführungsveranstaltung

Thomas Noll

Lehrstuhl für Informatik 2
(Software Modeling and Verification)



noll@cs.rwth-aachen.de

<http://www-i2.informatik.rwth-aachen.de/i2/algds13/>

9. Oktober 2013

1 Einführung

2 Termine

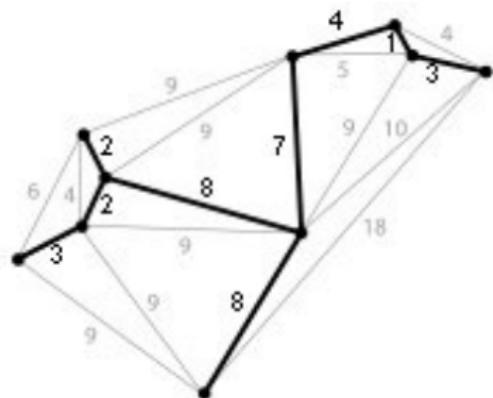
3 Algorithmen

4 Datenstrukturen

Thema des Proseminars

Weiterführung und Vertiefung diverser Themen der Vorlesung *Datenstrukturen und Algorithmen*

- Algorithmen:
 - neue Algorithmen
 - andere Komplexitätsmaße, ...
- Datenstrukturen:
 - Bäume
 - Hashing, ...
- Inhalt:
 - Problemstellung
 - Arbeitsweise des Algorithmus bzw. der Datenstruktur
 - Effizienzeigenschaften
 - Anwendungen



Ziele des Proseminars

- Selbstständiges Einarbeiten in ein neues Thema
- Literaturrecherche
- Darstellen des Inhalts in einer **wissenschaftlichen** Ausarbeitung
- Verständliches Präsentieren

Ausarbeitung

- Selbstständiges Verfassen einer **ca. 15-seitigen** Ausarbeitung
- **Vollständiges** Literaturverzeichnis
- Korrektes Zitieren
- **Plagiarismus:**
Die nicht gekennzeichnete Übernahme fremder Inhalte führt zum **sofortigen Ausschluss**.
- Schriftgröße **12pt**, übliche Seitenränder
- **Titelseite** mit Thema, Titel Proseminar, Semester, Name, Datum
- **Sprache** Deutsch oder Englisch
- **Korrekte Sprache** wird vorausgesetzt:
 ≥ 10 Fehler pro Seite \Rightarrow Abbruch der Korrektur

Vortrag

- 30-minütiger Vortrag
- Zielgruppengerechte Präsentation der Inhalte
- Übersichtliche Folien:
 - ≤ 15 Textzeilen
 - sinnvoller Einsatz von Farben
- Vortrag in Deutsch oder Englisch

1 Einführung

2 Termine

3 Algorithmen

4 Datenstrukturen

Verfahren

- Themenliste wurde/wird ausgehändigt
- Priorisierte Auswahl
- Wir bemühen uns (ohne Garantie) um ein “optimales” Matching
- Zuordnung der Themen bis **11 October** online
- Betreuer wurden bereits zugeordnet
- Themen 1 und 17 auf Englisch (zumindest Betreuung und Ausarbeitung), Rest nach Wahl

Rücktritt vom Proseminar

- Bis zu **drei Wochen** nach Einführung: ohne Folgen
- Danach: Fehlversuch

Einführung in die Literaturrecherche

- Einweisung in themenspezifische Literaturrecherche
- Dauer: ca. zwei Stunden
- Teilnahme **für BSc-Studierende verpflichtend**
- Bedarf bitte auf Themenblatt vermerken
- Termine:
 - Dienstag, 29.10., 12:00–14:00 Uhr
 - Donnerstag, 31.10., 14:45–16:45 Uhr
 - Montag, 04.11., 11:00–13:00 Uhr

Deadlines

Folgende Termine sind **verbindlich**:

- 31.10.2013: **letzte Rücktrittsmöglichkeit**
- 15.11.2013: Vorlage der Inhaltsübersicht
- 06.01.2014: erste Fassung der Ausarbeitung
- 13.01.2014: endgültige Fassung der Ausarbeitung
- 27.01.2014: erste Fassung der Folien
- 03.02.2014: endgültige Fassung der Folien
- 10./11.02.2014: Blockseminar

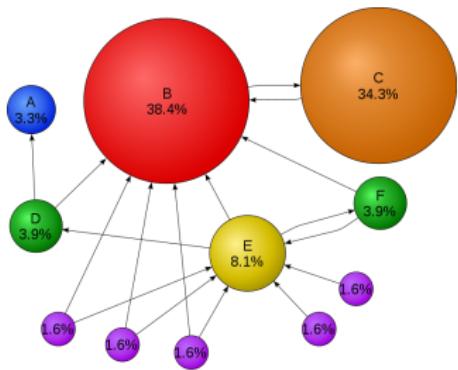
1 Einführung

2 Termine

3 Algorithmen

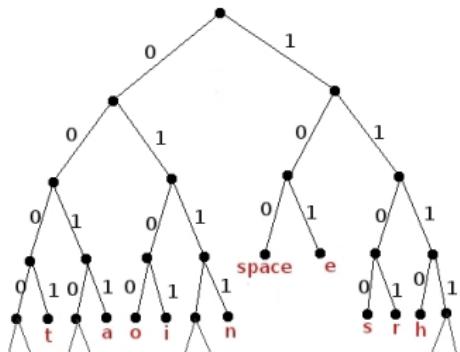
4 Datenstrukturen

1. Pagerank-Algorithmus [Englisch]



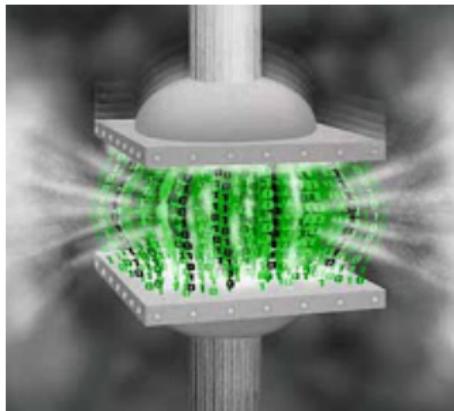
- Algorithmus zur Analyse verlinkter Webseiten
- Allgemeiner: beliebige Graphen
- Basis: Bestimmung der Wahrscheinlichkeit, dass zufälliges Benutzerverhalten auf die Seite führt
- Benutzung in Google-Suchmaschine

2. Huffman-Kodierung



- Ziel: Erstellung einer Binärkodierung mit minimaler mittlerer Wortlänge benötigt
- Basis: Wahrscheinlichkeitsverteilung der einzelnen Zeichen
- Ansatz: Verwendung eines vollen Binärbaums zur Darstellung des Codes

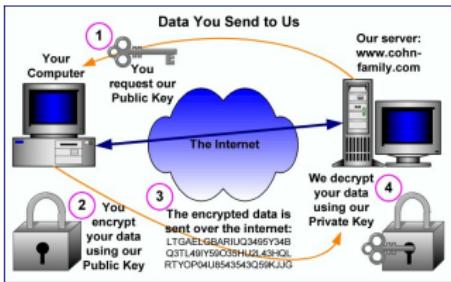
3. LZW-Datenkompression



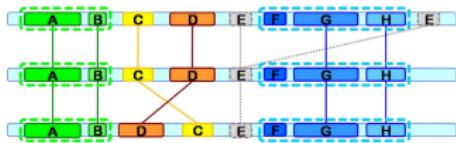
- Lempel-Ziv-Welch-Algorithmus
- Häufig bei Grafikformaten eingesetzt
- Verlustfrei
- Kompression mittels Wörterbüchern mit am häufigsten vorkommenden Zeichenketten
- Wörterbuch nicht zusätzlich gespeichert, sondern aus Datenstrom rekonstruiert

4. RSA-Kryptosystem

- Verfahren von Rivest, Shamir und Adleman (“public-key encryption”)
- Verwendung zur Verschlüsselung und digitalen Signatur
- Asymmetrisch: Schlüsselpaar privat/öffentlich
 - privat: zum Entschlüsseln oder Signieren
 - öffentlich: zur Verschlüsselung/Signaturprüfung
- Privater Schlüssel wird geheim gehalten und kann nur mit extrem hohem Aufwand aus dem öffentlichen Schlüssel berechnet werden

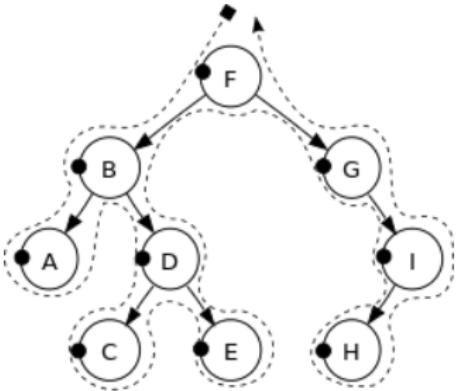


5. Longest Common Subsequence



- Definition: längste gemeinsame (nicht notwendigerweise zusammenhängende) Teilsequenz mehrerer Zeichenketten
- Anwendungen: diff, Bioinformatik
- NP-hart für beliebige Anzahl von Zeichenketten
- Quadratisch für zwei Zeichenketten (dynamische Programmierung)

6. Deutsch-Schorr-Waite Baumtraversierung



- Klassische Lösung: Rekursion/Stack
- Ziel: Vermeidung des zusätzlichen Speicheraufwands
- Ansatz: Pointerrotation in der Heapdarstellung

7. Problem der K kürzesten Pfade



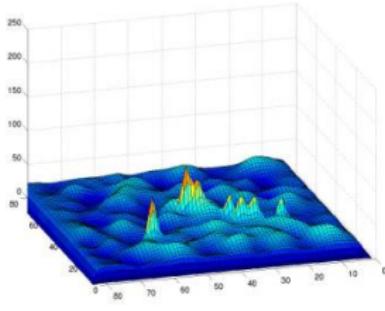
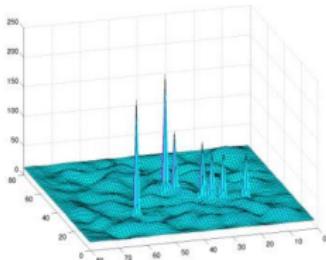
- Problem: Finden der K kürzesten Pfade zwischen Knotenpaar in einem gerichteten Graphen
- Ansatz: Verallgemeinerung der Bellman-Gleichung

8. Amortisierte Laufzeitanalyse



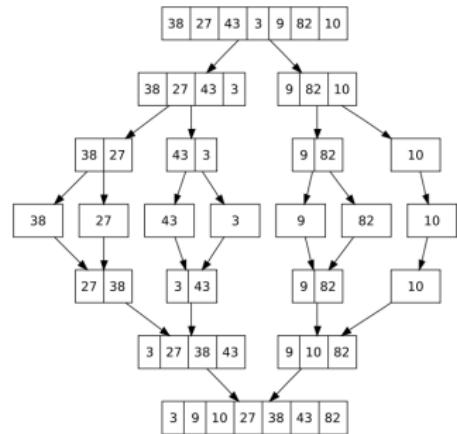
- Allgemeinen Laufzeitanalyse: maximale Kosten der einzelnen Schritte
- Amortisierte Laufzeitanalyse: Worst Case aller Operationen im gesamten Durchlauf des Algorithmus
- Verbesserung der oberen Schranke bei seltenem Auftreten teurer Operationen
- Idee: der Worst Case ändert den Zustand der Datenstruktur so ab, dass er nicht wiederholt auftreten kann (z.B. dynamische Arrays)
- Drei unterschiedliche Berechnungsmethoden (Aggregat-, Account-, Potentialfunktion-Methode)

9. Smoothed Analysis



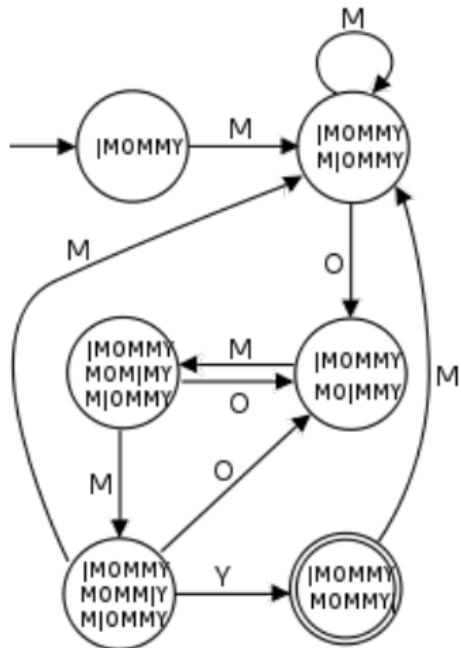
- Ziel: realistische Abschätzung des Laufzeitverhaltens von Algorithmen
- Kombination von Worst-Case- und Average-Case Analyse
- Annahme: “typische” Eingabewerte mit relativen geringen Abweichungen (“perturbation”)
- Liefert „geglättete“ Effizienzergebnisse

10. Externes Sortieren



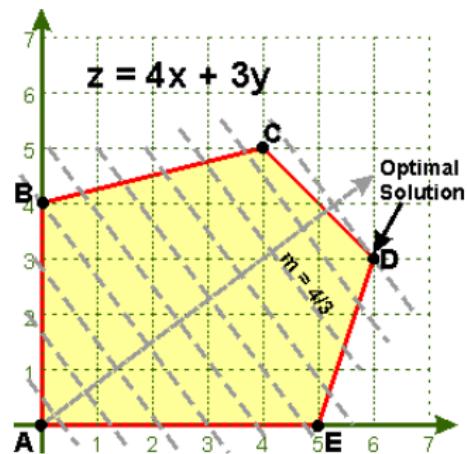
- Ziel: Sortierung großer Datenmengen unter Verwendung externer Speicher
- Typischer Ansatz: divide-and-conquer (sort-merge-Strategien)

11. String Matching



- Ziel: Suche von Vorkommen eines Strings in einem (großen) Text
- Komplexität des naiven Algorithmus:
 $|Text| \cdot |String|$
- Verbesserung durch Vorverarbeitung des Suchstrings

12. Lineare Optimierung (aka Lin. Progr.)

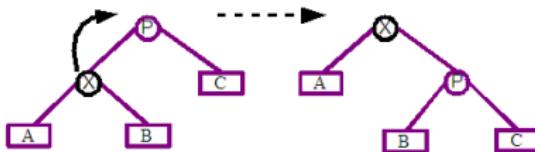


- Ziel: Optimierung linearer Zielfunktionen über einer Menge, die durch lineare (Un-)Gleichungen eingeschränkt ist
- Simplexverfahren
- Ellipsoid-Methode

- 1 Einführung
- 2 Termine
- 3 Algorithmen
- 4 Datenstrukturen

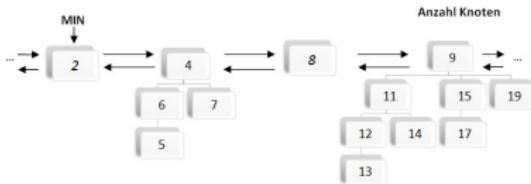
13. Splay Trees

Rotate Right



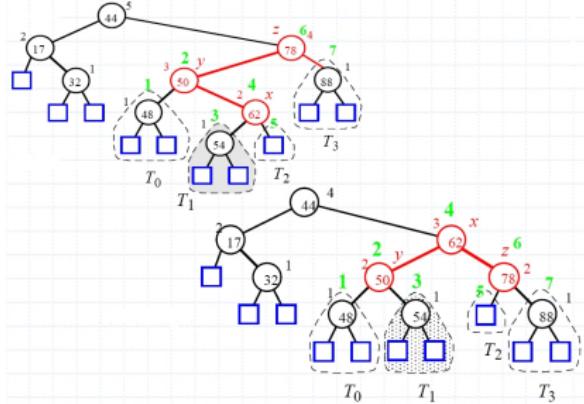
- Baumorganisierte Datenstruktur
- Effizient Einfügen, Suchen und Löschen von Elementen
- Grundlegende Operation: *splay* (Rotieren eines Elements bis zur Wurzel)

14. Fibonacci Heaps

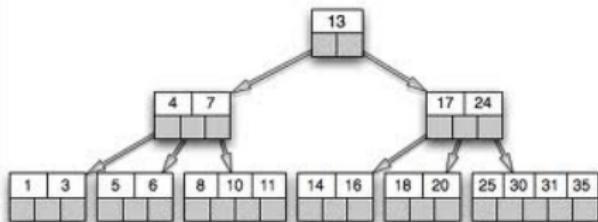


- Datenstruktur zur Implementierung einer Vorrangwarteschlange (*priority queue*)
- Liste von geordneten Bäumen
- Heap-Bedingung: Priorität jedes Knotens mindestens so groß wie Priorität seiner Kinder
- (Fast) alle Operationen haben amortisiert konstant Laufzeit

15. AVL-Bäume

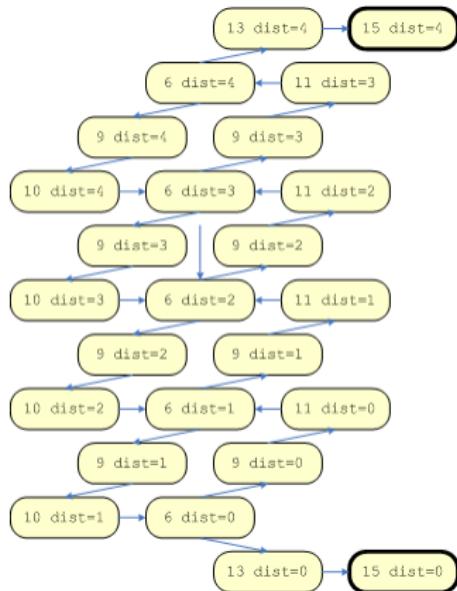


- Balancierter binärer Suchbaum
- Invariante: maximaler Höhenunterschied der Teilbäume jedes Knotens ist 1
- Worst-Case-Komplexität der üblichen Operationen: $O(\log n)$



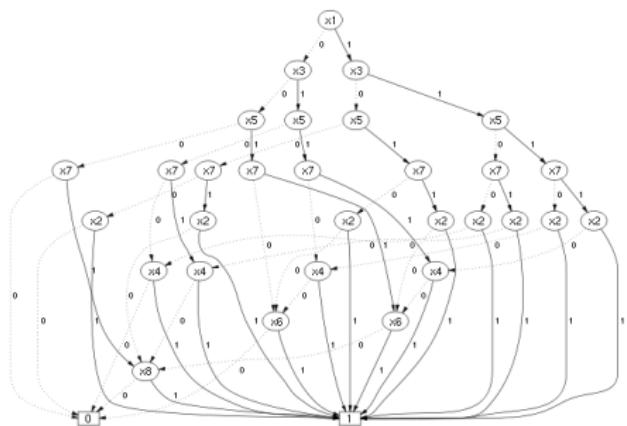
- Immer vollständig balanciert
- Besonderheit: Anzahl der Knotennachfolger variabel (mit Obergrenze)
- Einsatz vor allem in Datenbanken und Dateisystemen

17. Bitstate Hashing [Englisch]

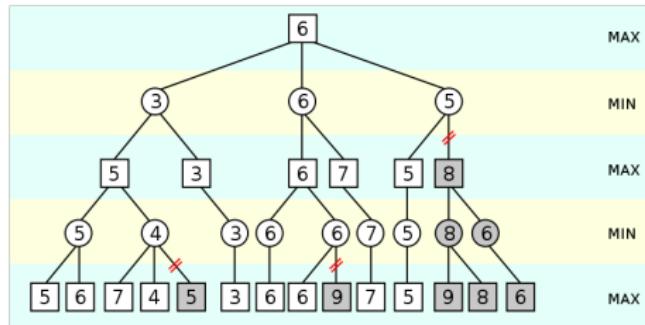


- Effiziente Darstellung von Zustandsräumen
- Problem: Erkennung von Zykeln
- Idee: Benutzung einer Hashfunktion, Speicherung von 0/1 im Hasharray
- Bis zu 98% Speicherersparnis im SPIN-Tool

18. Binäre Entscheidungsdiagramme

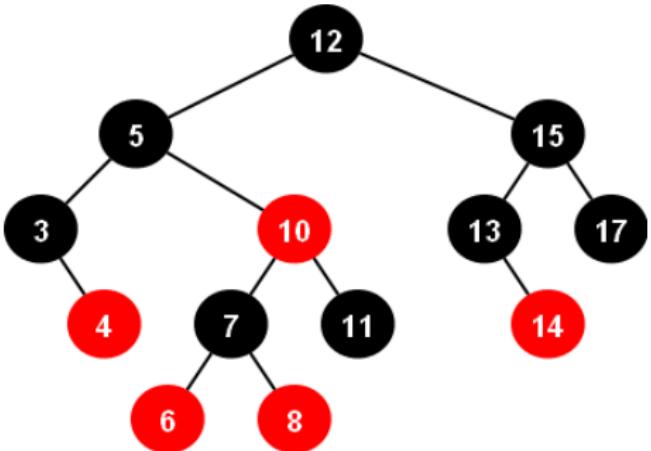


- Datenstruktur zur Darstellung Boolescher Funktionen
- Einsatz: Verifikation von (Hardware-)Systemen
- Problem: Finden geeigneter Variablenordnungen



- Darstellung von 2-Personen-Spielen mit abwechselnden Zügen (z.B. Schach, Go, Reversi, Dame, Mühle oder Vier gewinnt)
- Ziel: Bestimmung optimaler Strategien
- Minimax-Verfahren
- α - β -Pruning

20. Rot-Schwarz-Bäume



- Von binären Suchbäumen abgeleitet
- 5 charakterisierende Eigenschaften:
 - ① jeder Knoten ist rot oder schwarz
 - ② Wurzel ist schwarz.
 - ③ alle Blätter sind schwarz.
 - ④ beide Kinder eines roten Knotens sind schwarz
 - ⑤ jeder Pfad von einem Knoten zu seinen Blattknoten enthält die gleiche Anzahl schwarzer Knoten

⇒ annähernde Balancierung, effiziente Laufzeitschranken für Suchen/Einfügen/Löschen