



ulm university

universität

uulm



Seminarvorstellung WS 2017/2018 Institut für theoretische Informatik

Proseminar Algorithmen

Uwe Baier | 4.7.2017

Institut für theoretische Informatik



Uwe Schöning



Jacobo Torán



Enno Ohlebusch

Lehrveranstaltungen:

- ▶ Formale Grundlagen
- ▶ Algorithmen und Datenstrukturen
- ▶ Kryptologie
- ▶ Logik
- ▶ Berechenbarkeit und Komplexität
- ▶ Sequenzanalyse

Forschungsinteressen:

- ▶ Komplexitätstheorie
- ▶ Graphentheorie
- ▶ SAT
- ▶ Scheduling
- ▶ Kryptographie
- ▶ Datenkompression

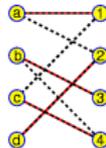
Proseminar Algorithmen

- ▶ Studiengänge
 - ▶ Informatik [B.Sc.]
 - ▶ Medieninformatik [B.Sc.]
 - ▶ Software-Engineering [B.Sc.]
- ▶ Themen
 - ▶ Kombinatorische Optimierung
 - ▶ Baumartige Strukturen
 - ▶ Stringanalyse
 - ▶ Komprimierungstechniken
- ▶ Vorkenntnisse
 - ▶ Algorithmen und Datenstrukturen (optional)
- ▶ Dozenten
 - ▶ Enno Ohlebusch
 - ▶ Uwe Baier

Kombinatorische Optimierung

► Perfektes Matching mit minimalem Gewicht:

Gegeben sei ein gewichteter bipartiter Graph, finde ein PM mit minimalem Gewicht

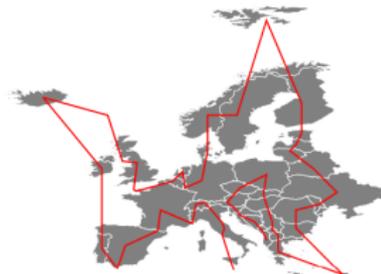


► Traveling Salesman Approximation:

Gegeben sei ein (vollständiger) Graph, in dem die Dreiecksungleichung erfüllt ist:

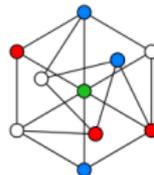
$$w(u, v) \leq w(u, x) + w(x, v)$$

Finde eine gute Approximation für eine optimale Rundtour



► Knotenfärbung:

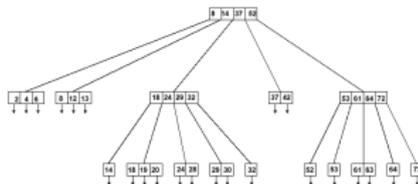
Gegeben sei ein Graph, finde eine Färbung der Knoten, sodass keine zwei Nachbarknoten dieselbe Farbe besitzen.



Baumartige Strukturen

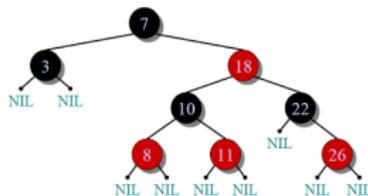
► B-Bäume:

sortierter Baum mit m
Einträgen pro Knoten und
Verzweigungsgrad $m + 1$



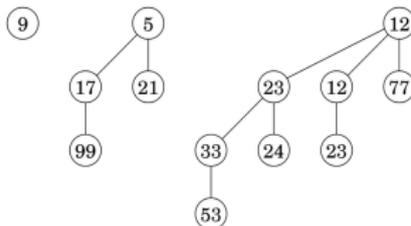
► Rot-Schwarz-Bäume:

Durch Färbungsregeln
balancierter Suchbaum



► Binomial Heaps:

Heap, dessen Struktur
durch Binomialkoeffizient
bestimmt wird



Stringanalyse

▶ Rank und Select auf Bitvektoren:

Gegeben sei ein Bitvektor $B[1..n]$, bestimme in $O(1)$

- ▶ $\text{rank}_B(1, i) = \# \text{Anzahl gesetzter Bits in } B[1..i]$
- ▶ $\text{select}_B(1, i) = \# \text{Position des } i\text{-ten gesetzten Bits in } B$

▶ Wavelet Tree:

Gegeben sei ein String $S[1..n]$ über Alphabet Σ , bestimme in $O(\log |\Sigma|)$

- ▶ $\text{rank}_S(c, i) = \# \text{Anzahl Vorkommen des Buchstaben } c \text{ in } S[1..i]$
- ▶ $\text{select}_S(c, i) = \# \text{Position des } i\text{-ten Vorkommens von } c \text{ in } S$

i	1	2	3	4	5	6	7	8	9	10
$B[i]$	1	0	0	1	0	0	0	1	1	0

$\text{rank}_B(1, 5) = 2$

$\text{select}_B(1, 3) = 8$

Komprimierungstechniken

► **Huffman-Kodierung und arithmetische Kodierung:**

Wie klein kann ein Text durch Kodierung einzelner Buchstaben minimal werden?
Finde optimalen Code für gegebenen Text

$$\begin{array}{l} a \rightarrow 0; \quad b \rightarrow 10; \quad c \rightarrow 11 \\ \text{abaabc} \quad \rightarrow \quad 0_1 10_1 0_1 0_1 10_1 11 \end{array}$$

► **LZ77 und LZ78:**

Kodierung eines Textes mittels Referenzen auf Repeats

t e x t i t e x t t e x t

► **Burrows-Wheeler-Transformation:**

Permutation der Buchstaben eines Textes

⇒ Einsatz in Datenkompression und Sequenzanalyse

Proseminar Algorithmen: Ablauf

- ▶ Proseminar, 4 ECTS-Punkte, Ausarbeitung + Vortrag + aktive Teilnahme
- ▶ Blockveranstaltung (Vorträge gegen Ende der Vorlesungszeit)
- ▶ Vorbereitungen zu Beginn des Wintersemesters, Informationen dazu folgen auf der Homepage
- ▶ Vorlesung Algorithmen und Datenstrukturen empfohlen