

Klausurvorbereitung – Algorithmen II

Michael Axtmann – michael.axtmann@kit.edu

http://algo2.iti.kit.edu/AlgorithmenII_WS16.php

Institut für Theoretische Informatik - Algorithmik II

```
    result = current_weight;
    return true;
}

for( EdgeID eid = graph.edgeBegin( current ); eid != graph.edgeEnd( current ); ++eid ){
    const Edge & edge = graph.getEdge( eid );
    COUNTING( statistic_data.inc( DijkstraStatisticData::TOUCHED_EDGES ); )
    if( edge.forward ){
        COUNTING( statistic_data.inc( DijkstraStatisticData::RELAXED_EDGES ); )
        weight new_weight = edge.weight + current_weight;
        GUARANTEE( new_weight >= current_weight, std::runtime_error, "Weight overflow detected." );
        if( !priority_queue.isReached( edge.target ) ){
            COUNTING( statistic_data.inc( DijkstraStatisticData::SUCCESSFULLY_RELAXED_EDGES ); )
            COUNTING( statistic_data.inc( DijkstraStatisticData::REACHED_NODES ); )
            priority_queue.push( edge.target, new_weight );
        } else {
            if( priority_queue.getCurrentKey( edge.target ) > new_weight ){
                COUNTING( statistic_data.inc( DijkstraStatisticData::SUCCESSFULLY_RELAXED_NODES ); )
                priority_queue.decreaseKey( edge.target, new_weight );
            }
        }
    }
}
```

- Verwandte und vertiefende Vorlesungen
- Forschung am Lehrstuhl “Algorithmik II”
 - Aktuelle Forschung
 - Abschlussarbeiten
- Informationen zur Klausur
- Wiederholung der Vorlesungsinhalte
(was war Inhalt?)

Vorlesung	LP	Wahl/Vert.	Sem.	Details
Algorithmen für planare Graphen	5	W	SS	B
Algorithmen für Routenplanung	5	W/AT	SS	M
Algorithm Engineering	5	W/AT	SS	M
Parallele Algorithmen	5	W/AT/PV	WS	M
Text-Indexierung	5	W/AT	WS	M
Fortgeschrittene Datenstrukturen	5	W/AT/TG	SS	M
Graphpartitionierung und Graph-clustern in Theorie und Praxis	5	W/AT/TG	SS	M
SAT Solving in der Praxis	5	TG	SS	M
Proofs from THE BOOK	3	Seminar	SS17	M

W: Wahlbereich Informatik

AT: Wahl Algorithmentechnik

PV: Wahl Parallelverarbeitung

TG: Wahl Theoretische Grundlagen

Lecture

Boolean Satisfiability (SAT) is **the** NP-complete problem

- Learn how SAT solvers work
- and how to use to solve other NP problems
- 2SWS Lectures + 1 SWS Exercises (5 ECTS)
- starting 24.4, 14:00, room 236, **in English**

Open Student Topics:

- Parallel Cooperative Local Search SAT Solving
- Parallelize Stålmarcks SAT Algorithm
- Parallel Longest Path Search
- Preprocessing For Incremental SAT Solving

Contact: Dr. Tomas Balyo, room 210

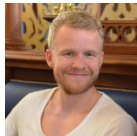
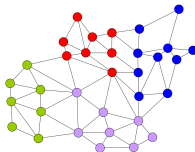
Vorlesung/Blockveranstaltung

Partition graph into k disjoint blocks $V_1, \dots, V_k$ s.t.

- total node weight of each block $\leq \frac{1+\epsilon}{k}$ total node weight
- objective as small as possible

Dates:

- 6.6-14.6 – je 2VLen täglich
- bei Bedarf: 24.7-27.7



Open Student Topics:

- FM-minimization of communication volume and other objectives
- partitioning of social networks with replicated nodes
- shared-memory parallel linear time graph drawing
- ... **many** more (also in cooperation with TAA in [Vienna](#))

Lecture

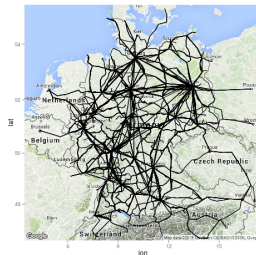
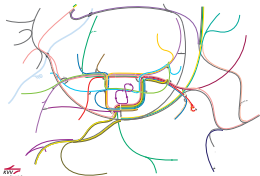
- Text Indexing (winter term 2017/2018)
 - Learn about design, construction, and applications of advanced text indexes
 - Compressed text indexes (FM-Index, CSA, CST)
 - Indexes for document collections (web pages, genome collections)

Open Student Topics:

- Compression of phylogenetic tree inferences
- Efficient navigation in compressed trees
- Space-efficient representation of triangulations
- Grammar-compressed text indexes

Contact: Dr. Simon Gog, room 206

Public Transit Routing



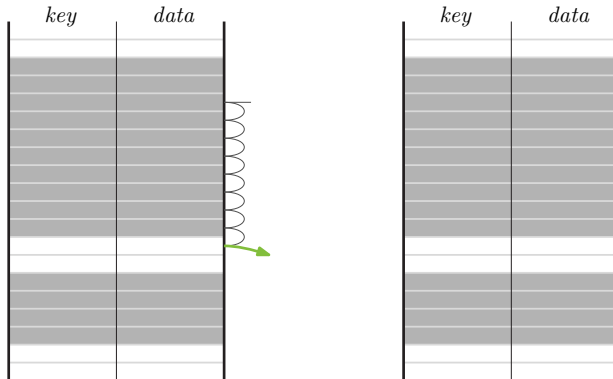
- Alltägliches Problem
- Techniken aus der Routenplanung auf Straßen kaum anwendbar
- Grundproblem deutlich schwieriger
 - Komplexere Netzwerke
 - Zeitabhängig, multikriteriell, range queries, ...
- Neue Sichtweise führt zu neuen Algorithmen

KIT
Karlsruher Institut für Technologie

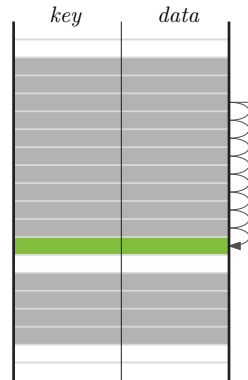
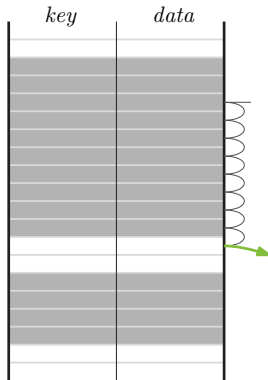
[illegible]

KIT
Karlsruher Institut für Technologie

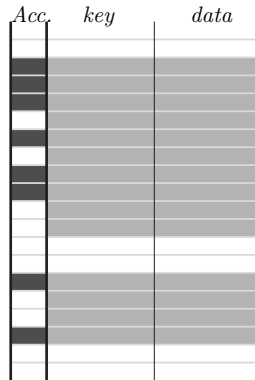
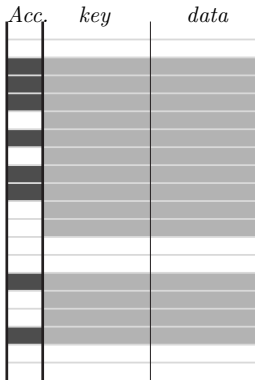
Hash Tables - linear probing



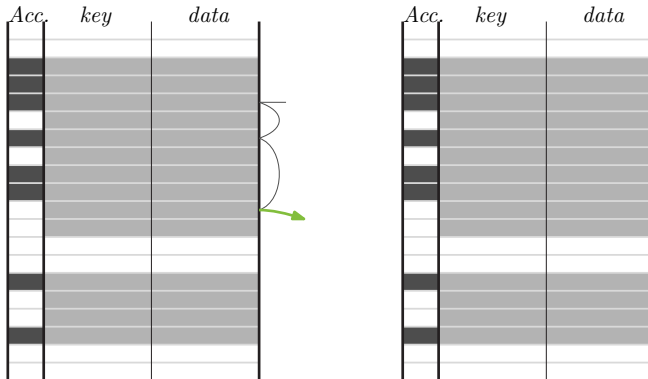
Hash Tables - linear probing



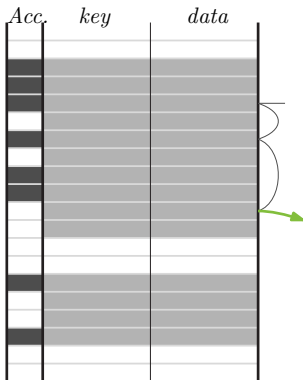
Hash Tables - hopscotch hashing



Hash Tables - hopscotch hashing



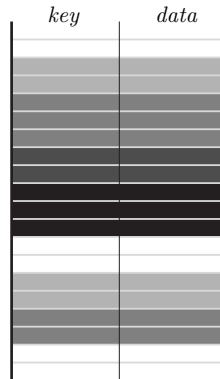
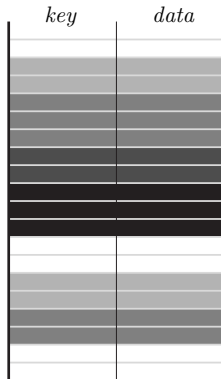
Hash Tables - hopscotch hashing



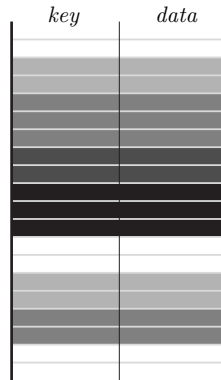
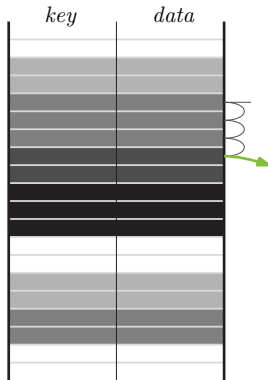
KIT
Karlsruher Institut für Technologie

[illegible]

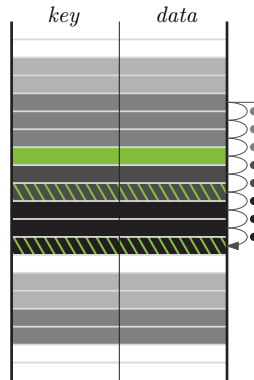
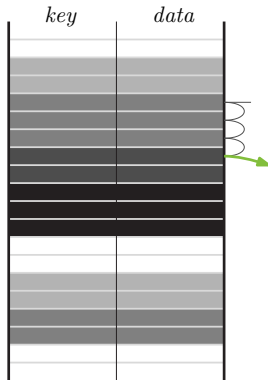
Hash Tables - robin hood



Hash Tables - robin hood

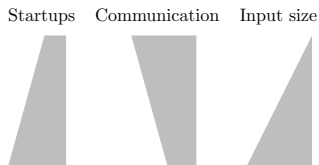
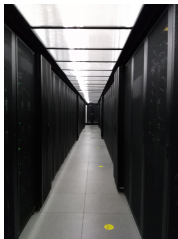
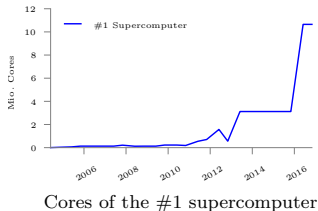


Hash Tables - robin hood



Massively Parallel Sorting

Data source: TOP500 November 2014



Issues:

- Message startups expensive
- Robustness
 - Input size
 - Repeated keys
 - Irregular distr. of keys
- Network topology

We provide:

- Write programs in C++ and MPI
- Runs on more than 500k cores

Vorlesung/Übung am Mittwoch fällt aus

- Vorlesung hat den ganzen Stoff abgearbeitet
- Übung abgeschlossen
 - alles schon klar?
 - noch nicht angeschaut?
- Falls es Fragen gibt: eMail, Forum, Sprechstunde

Vorlesung/Übung am Mittwoch fällt aus

- Vorlesung hat den ganzen Stoff abgearbeitet
- Übung abgeschlossen
 - alles schon klar?
 - noch nicht angeschaut?
- Falls es Fragen gibt: **eMail**, **Forum**, **Sprechstunde**

- **Termin:** Mittwoch, 22. Februar (14:30 Uhr)
- **Dauer:** 2 Stunden
- **Hilfsmittel:** 1 handbeschriebenes DIN A4 Blatt (zweiseitig)

- **Voraussichtlicher Inhalt:** 7 Aufgaben
 - Aufgabe 1: Kleinaufgaben (über verschiedene Kapitel)
 - Aufgabe 2 – 7: große Aufgaben (über je ein Kapitel)

Aufgabentypen:

- **Rechenaufgaben**
(Algorithmus aus der Vorlesung auf gegebener Instanz ausführen)
- **Entwurfsaufgaben**
(Lösungsansatz zu gegebenem Problem beschreiben)
- **Analyseaufgaben**
(Beweise, Laufzeiten, ...)

Ausgeschlossene Themen aus der Vorlesung

- **Folien mit “Chilli”** ()

- **Übungsblätter** und **Übungen** wiederholen!
(bzw. zum ersten Mal anschauen)
- Alle Aufgaben
 - einmal **lösen**, oder zumindest **verstanden haben**!
- **Aussagen** / **Definitionen** merken
→ dürfen in Klausur verwendet werden!