# Dagstuhl Seminar

# No. 02091, Report No. 335

24.02.2002 – 01.03.2002

# Data Structures


Susanne Albers (Univ. Freiburg, Germany)

Robert Sedgewick (Princeton Univ., USA)

Peter Widmayer (ETH Zürich, Switzerland)

The area of Data Structures continues to be an important and vibrant aspect of computer science. The topic is an essential component in the algorithmic solution of many problems. Although data structures have been studied for four decades, there is still a large research community working on exciting and challenging problems. The Sixth Dagstuhl Seminar on Data Structures was attended by 59 people and hence it was larger than all previous meetings. Attendees came from 13 different countries and included many young colleagues. About a third of the participants were attending the seminar for the first time, bringing new ideas and points of view.

There were 40 workshop presentations and, despite of the high attendance, there was sufficient time for scientific discussions and research in teams. The presentations addressed classical data structuring problems as well as new problems arising in important applications. Many interesting results were presented on classical issues such as dictionaries, ordered lists, ordinary search trees, finger trees, B-trees and priority queues. A number of lectures considered classical graph problems. Several presentations investigated data structuring problems in computational geometry, in particular geometric problems with moving objects. With respect to external memory algorithms, several talks presented cache oblivious solutions that need no knowledge of the exact parameters of the memory hierarchy. Last but not least, there were several contributions investigating data structure problems in specific application areas such as Networks, Parallel Computing and Database Systems.

As usual there were fruitful and stimulating discussions and we hope that, with respect to the scientific results, this seminar will be as successful as the one in 2000.

**Thanks:** The abstracts of the 40 talks were compiled by David Taylor. The organizers would like to thank all participants and especially the team of Schloß Dagstuhl for helping to make the workshop a success. The warm atmosphere of the Schloß, as always, supported discussions and the informal exchange of ideas!

# Contents

# 1 Abstracts

## Monday, February 25

### One-Probe Search

Rasmus Pagh

Joint work with Anna Östlin.

We consider dictionaries that perform lookups by probing a *single word* of memory, knowing only the size of the data structure. We describe a randomized dictionary where a lookup returns the correct answer with probability $1 - \epsilon$, and otherwise returns "don't know". The lookup procedure uses an expander graph to select the memory location to probe. Recent explicit expander constructions are shown to yield space usage much smaller than what would be required using a deterministic lookup procedure. Our data structure supports efficient *deterministic* updates, exhibiting new probabilistic guarantees on dictionary running time.

### Algorithms for Path-Based Placement of Inspection Stations on Transportation Networks

S. S. Ravi

This work was carried out jointly with Professor Daniel Rosenkrantz (Department of Computer Science, SUNY Albany) and Professor Giri Tayi (School of Business, SUNY Albany).

Placement of inspection stations is a common task in the transportation of hazardous materials. The purpose of inspection is to ensure that the vehicle carrying hazardous materials is in a safe condition to continue along its path.
We consider two categories of problems involving the placement of inspection stations. The first category deals with the selection of inspection stations along a given path. The second category considers the simultaneous selection of a path and the placement of inspection stations along that path. We consider these formulations under several optimization objectives. We present efficient

algorithms for some formulations as well as complexity and approximation results for some others.

## Top-Down Analysis of Path Compression

Raimund Seidel

Let $u$ be a node in a forest $F$ and let $v$ be some ancestor of $u$. Compressing the path from $u$ to $v$ means changing $F$ so that every node on that path except for $v$ gets $v$ as its parent. The cost of such a path compression is the number of nodes that get a new parent. What is the maximum cumulative cost of a sequence of $m$ path compressions in a forest with $n$ nodes? This question is of interest in the analysis of some algorithms for the so-called union-find problem. More than 25 years ago Tarjan showed tight upper bounds of $O(m \log_{\lceil 1+m/n \rceil} n)$ for the general case and $O(m\alpha(m,n))$ for the case when $F$ is in some sense balanced.
We rederive these bounds using a very simple top-down recursive analysis approach. Our main tool is the following lemma: Let the nodes of a forest $F$ be partitioned in an upper part $U$ and a lower part $L$, so that the parent of every node in $U$ is also in $U$. Let $F_U$ and $F_L$ be the subforests of $F$ induced by $U$ and $L$, respectively. For every sequence $C$ of path compressions in $F$ there are sequences $C_U$ and $C_L$ of path compressions in $F_U$ and $F_L$, respectively, so that $|C| = |C_U| + |C_L|$ and

$$cost(C) \leq cost(C_U) + cost(C_L) + |L| + |C_U| \, .$$

Here $|C|$ denotes the length of the sequence $C$, i.e. the number of path compressions in the sequence.

## Labeling Schemes for Trees

Theis Rauhe

Joint work with Stephen Alstrup.

We consider the problem of assigning short labels to the nodes of a tree such that adjacency between any pair of nodes can be tested based on the labels of the two nodes alone. We show that there exists a labeling scheme with labels of size $logn + O(log^*n)$. The result implies the existence of an induced-universal graph for trees with size $n2^O(log^*n)$ nodes improving a bound by Chung of $O(nlogn)$.

## Lower Bounds for Ancestor Labeling

Stephen Alstrup

This is joint work with Theis Rauhe.

Given a tree, a labeling scheme assigns a label, which is a binary string, to each node. Given the labels of any two nodes $u$ and $v$, it can be determined, if a certain property holds for $u$ and $v$, alone from these labels.
For trees of size $n$, we show that any labeling scheme for testing ancestor relation uses labels of size $\log n + \Omega(\log \log n)$ to the nodes.

## Routing in Large Transportation Networks

Madhav Marathe

This is joint work with Chris Barrett, Keith Bisset, Riko Jacob, and Goran Konjevod.

In many path finding problems arising in diverse areas, such as transportation science, Web search, Database queries and VLSI design certain patterns of edge/vertex labels in the labeled graph being traversed are allowed/preferred, while others are disallowed. Thus, the feasibility of a path is determined by (i) its length (or cost) under well known measures on graphs such as distance, and (ii) its associated label. We model the set of acceptable label patterns as a formal language. For example, in transportation systems with mode options, formal languages can be used to represent modal preferences of travelers. Motivated by these applications, we consider the following prototypical class of problems:

Formal Language Constrained Shortest/Simple Paths: Given a labeled weighted (directed or undirected) graph $G$, a source and a destination $s$ and $d$, and a formal language $\mathcal{L}$ such as regular, context free, etc) find a shortest (or simple) path from $s$ to $d$ such that the label of the path is in $\mathcal{L}$.

In this talk I will outline a number of theoretical results for this and related problems focusing on the efficient implementations. In order to evaluate our ideas in a practical setting, I will report computational experience on the performance of the proposed algorithms. This experience was gained in the context of the ongoing TRANSIMS (Transportation Analysis and Simulation System) project at LANL.

## An Efficient Quasidictionary

### Torben Hagerup

Joint work with Rajeev Raman.

We define a *quasidictionary* to be a data structure that supports the following operations: check-in($v$) inserts a data item $v$ and returns a positive integer tag to be used in future references to $v$; check-out($x$) deletes the data item with tag $x$; access($x$) inspects and/or modifies the data item with tag $x$. A quasidictionary is similar to a dictionary, the difference being that the names identifying data items are chosen by the data structure rather than by its user. We describe a deterministic quasidictionary that executes the operations check-in and access in constant time and check-out in constant amortized time, works in linear space, and uses only tags bounded by the maximum number of data items stored simultaneously in the quasidictionary since it was last empty.

## A Certifying Linear Time Recognition Algorithm for Interval Graphs

### Kurt Mehlhorn

Joint work with Dieter Kratsch, Ross McConnell and Jeremy Spinrad.

We present a linear-time recognition algorithm for interval graphs that outputs an interval model if the input graph is an interval graph (as is done by all known linear-time recognition algorithms for interval graphs) and that outputs an asteroidal triple or a chordless cycle of length at least four if the input graph is not an interval graph, i.e. it provides a certificate of non-membership (as a new feature).

## Three-Dimensional Layers of Maxima

### Adam L. Buchsbaum

This is joint work with Michael T. Goodrich (UC Irvine).

Given $n$ points in some space, one point, $p$, *dominates* another, $q$, if each coordinate of $p$ is greater than that of $q$. A point not dominated by any other point is in *layer* 1; removing the layer-1 points, the remaining points not dominated by other points are in layer 2; and so on. The *layers of maxima* problem is to determine the layer of each point. We present a new algorithm that solves this problem for points in $\Re^3$ in $O(n \log n)$ time and $O(n \log n / \log \log n)$ space. (A previous $O(n \log n)$-time solution, due to Atallah, Goodrich, and Ramaiyer [1994], has technical flaws.) Our algorithm uses a new extension of dynamic fractional cascading that alleviates the degree constraint on the control graph, which might be of independent interest.

## Tuesday, February 26

## Kinetic Heaps

Haim Kaplan

We describe several implementations of the *kinetic heap*, a heap (priority queue) in which the key of each item, instead of being fixed, is a linear function of time. The kinetic heap is a simple example of a kinetic data structure of the kind considered by Basch, Guibas, and Hershberger. Kinetic heaps have many applications in computational geometry, and previous implementations were designed to address these applications. We describe an additional application, to broadcast scheduling. Each of our kinetic heap implementations improves on previous implementations by being simpler or asymptotically faster for some or all applications.

## STAR-Tree: A Data Structure for Handling Moving Points

Pankaj K. Agarwal

We present a new technique called STAR-tree, based on R*-tree, for indexing a set of moving points so that various queries, including range queries, time-slice queries, and nearest-neighbor queries, can be answered efficiently. A novel feature of the index is that it is self-adjusting in the sense that it re-organizes itself locally whenever its query performance deteriorates. The index provides tradeoffs between storage and query performance and between time spent in updating the index and in answering queries. We present detailed performance studies and compare our methods with the existing ones under a varying type of data sets and queries. Our experiments show that the index proposed here performs considerably better than the previously known ones.

## Computing Large Planar Regions in Terrains

Michiel Smid

This is joint work with Rahul Ray, Ulrich Wendt, and Katharina Lange.

We consider the problem of computing the largest region in a terrain that is approximately contained in some two-dimensional plane. We reduce this problem to the following one. Given an embedding of a degree-3 graph $G$ on the unit sphere, whose vertices are weighted, compute a connected subgraph of maximum weight that is contained in some spherical disk of a fixed radius. We given an algorithm that solves this problem in $O(n^2 \log n (\log\log n)^3)$ time, where $n$ denotes the number of vertices of $G$ or, alternatively, the number of faces of the terrain. We also give a heuristic that can be used to compute sufficiently large regions in a terrain that are approximately planar. We present a web-based implementation of this heuristic, and show some results for terrains representing three-dimensional (topographical) images of fracture surfaces of metals obtained by confocal laser scanning microscopy.

## Succinct Indexable Dictionaries with Applications to Encoding $k$-ary Trees and Multisets

Rajeev Raman

Joint work with Venkatesh Raman, IMSc and S. Srinivasa Rao, Leicester.

We consider the *indexable dictionary* problem which consists in storing a set $S \subseteq \{0, \ldots, m-1\}$ for some integer $m$, while supporting the operations of $rank(x)$, which returns the number of elements in $S$ that are less than $x$ if $x \in S$, and $-1$ otherwise; and $select(i)$ which returns the $i$-th smallest element in $S$.
We give a structure that supports both operations in $O(1)$ time on the RAM model and requires $\mathcal{B}(n,m) + o(n) + O(\lg\lg m)$ bits to store a set of size $n$, where $\mathcal{B}(n,m) = \lceil \lg \binom{m}{n} \rceil$ is the minimum number of bits required to store any $n$-element subset from a universe of size $m$. Previous dictionaries taking this space only supported (yes/no) membership queries in $O(1)$ time. In the cell probe model we can remove the $O(\lg\lg m)$ additive term in the space bound, answering a question raised by Fich and Miltersen, and Pagh. We present several applications of our dictionary structure including:

- an information-theoretically optimal representation for *k-ary cardinal trees* (aka $k$-ary tries) that uses $\mathcal{C}(n,k) + o(n + \lg k)$ bits to store a $k$-ary tree with $n$ nodes and can support parent, $i$-th child, child labeled $i$, and the degree of a node in constant time, where $\mathcal{C}(n,k)$ is the minimum number of bits to store any $n$-node $k$-ary tree. Previous space efficient representations for cardinal $k$-ary trees required $\mathcal{C}(n,k) + \Omega(n)$ bits;

- a representation for multisets where (appropriate generalizations of) the *select* and *rank* operations can be supported in $O(1)$ time. Our structure uses $\mathcal{B}(n, m+n) + o(n) + O(\lg\lg m)$ bits to represent a multiset of size $n$ from an $m$ element set; the first term is the minimum number of bits required to represent such a multiset.

We also highlight two other results that may prove to be useful subroutines in developing succinct data structures:

- we give a representation of a sequence of $m$ bits, of which $n$ are 1s, that uses $\mathcal{B}(n,m) + o(n)$ bits whenever $m = O(n\sqrt{\lg n})$, and answers the following queries in constant time: given a position $i$ in the bit-vector, to report the number of 0s (or 1s) before position $i$ in the bit-vector and given a number $i$, to return the position of the $i$-th 0 (or 1) in the bit-vector. This generalizes results by Clark and by Pagh.

- we give a representation of a sequence of $n$ non-negative (or strictly positive) integers summing up to an integer $m$ in $\mathcal{B}(n, m+n)+o(n)$ bits (or $\mathcal{B}(n,m)+o(n)$ bits respectively), to support partial sum queries in constant time; in each case the first term is the minimum number of bits required to represent the partial sums. This is more space-efficient than related results by Grossi and Vitter, Tarjan and Yao, Pagh, and Hagerup and Tholey.

## So Many Children with So Little Potential

Kim S. Larsen

This is joint work with Lars Jacobsen and Morten N. Nielsen.

In amortized analysis of data structures, it is standard to assume that initially the structure is empty. Usually, results cannot be established otherwise. We investigate the possibilities of establishing such results for initially non-empty multi-way trees.

## Load Balancing by Iterated Bisection: Nonuniform, Center-Oriented Distributions

Martin Dietzfelbinger

Let $\mathcal{P}$ be a set of problem instances (finite sequence of numbers, finite set of points, segment of space, etc.). Each $p \in \mathcal{P}$ is assumed to have a weight $w(p) > 0$. We assume that each problem instance can be split randomly into two pieces $p_1, p_2$ so that $w(p_1) + w(p_2) = w(p)$. The distribution $\mu_p$ of $w(p_1)/w(p)$ in $[0, 1]$ is symmetric (w.l.o.g.). Starting with some instance $p_0$, iterate splitting fragments, until (after $N - 1$ cuts) there are $N$ pieces $q_1, \ldots, q_N \in \mathcal{P}$. Bischof, Schickinger, and Steger have shown that if always the heaviest piece is split and for all $p$ $\mu_p$ is the uniform distribution then $\max\{w(q_1), \ldots, w(q_N)\}$ is close to $2\frac{w(p_0)}{N}$ with high probability. — We show that for a much wider class of distributions which are "center-oriented" in a sense made precise in the talk, the maximal size of a fragment is bounded by $(2 + \varepsilon)\frac{w(p_0)}{N}$ with high probability.

## Polynomial Time Algorithms for 2-Edge-Connectivity Augmentation Problems

Guido Proietti

Joint work with Anna Galluccio, Istituto di Analisi dei Sistemi ed Informatica, CNR, Roma, Italy.

Given a graph $G$ with $n$ vertices and $m$ edges, the 2-edge-connectivity augmentation problem is that of finding a minimum weight set of edges of $G$ to be added to a spanning subgraph $H$ of $G$ to make it 2-edge-connected. The problem is NP-hard, in general, and approximable. In this talk we show that in a special case, namely when $H$ is a depth-first search tree of $G$, then the problem can be polynomially solved. We also provide an efficient algorithm to solve this special case which runs in $\mathcal{O}(M \cdot \alpha(M, n))$ time, where $\alpha$ is the classic inverse of the Ackermann's function and $M = m \cdot \alpha(m, n)$. Applications to the problem of recovering the 2-edge-connectivity of a communication network after a link failure are shown.

## The Index Fabric: Properties and Challenges

Gisli R. Hjaltason

The Index Fabric is a balanced, hierarchical indexing structure based on the Patricia trie, that is suitable for large collections of long keys. To represent the Patricia trie on external storage, the trie is partitioned into connected subtries that fit into fixed-size disk blocks. Naturally, this potentially leads to a very unbalanced hierarchy of blocks. The novel aspect of the Index Fabric is that this unbalanced "vertical" hierarchy is augmented by a "horizontal" hierarchy of successively smaller layers of tries. Search for a given key proceeds from the smallest layer (that fits into a single block) to the largest (that contains the full Patricia trie), accessing a single block in each layer. Data represented in the Index Fabric is encoded using semantic tags embedded in the indexed keys, so multiple search paths can be present in the same index. Furthermore, relationships can be represented in the index by building composite keys with elements from different data elements, thus allowing search over a relationship without requiring join operations.
We briefly described two challenges that we have faced in reaching a practical implementation of the Index Fabric, namely that of obtaining a small index structure and that of concurrency control, and sketched our solution to the former problem. There are multiple aspects to the solution, which together yielded a compact subtrie representation in each block with high storage utilization but without undue sacrifice in CPU performance. The result was that the index typically occupies a third of the space compared to our initial implementation, with only about 1 to 3 byte overhead per key value (before taking the storage utilization into account, which typically approaches 70%).

## Approximation Algorithms for Fractional Covering and Structured Concave Optimization Problems

Klaus Jansen

Joint work with Lorant Porkolab (PricewaterhouseCoopers London).

We study resource constrained scheduling problems where the objective is to compute feasible preemptive schedules minimizing the makespan and using no more resources than what are available. We present approximation schemes along with some inapproximibility results showing how the approximability of the problem changes in terms of the number of resources. The results are based on linear programming formulations (though with exponentially many variables) and some interesting connections between resource constrained scheduling and (multi-dimensional, multiple-choice, and cardinality constrained) variants of the classical knapsack problem. In order to prove the results we generalize a method by Grigoriadis et al. for the structured concave optimization problem to the case with weak approximate block solvers (i.e. with only constant, logarithmic, or even worse approximation ratios).

## Wednesday, February 27

### Frequency Counts on Limited Memory

Alex Lopez-Ortiz

Joint work with E. Demaine (MIT) and Ian Munro (University of Waterloo).

We consider the problem of finding in real time IP destination frequencies of a packet stream, This is typically achieved using a combination of router facilities, dedicated hardware and a general computer. Packets flow in the Internet at a rate of one every eight nanoseconds on the average, which severely limits the amount of computation per packet..
We present a method for deterministically finding the top $k$ elements so long as they all appear with probability $p > 1/m$. We then propose a multiple round algorithm that finds the top $k$ packets so long as they appear with frequency $p > \log n/\sqrt{mn}$.

### Average-Case Analysis of Parallel Shortest-Paths Algorithms

Ulrich Meyer

We study the average-case complexity of the parallel single-source shortest-path problem, assuming arbitrary directed graphs with $n$ nodes, $m$ edges, and independent random edge weights uniformly distributed in $[0, 1]$. We provide a new bucket-based parallel SSSP algorithm that runs in $T = \mathcal{O}(\log^2 n \cdot \min_i\{2^i \cdot \mathcal{L} + |V_i|\})$ average-case time using $\mathcal{O}(n + m + T)$ work on a PRAM where $\mathcal{L}$ denotes the maximum shortest-path weight and $|V_i|$ is the number of graph vertices with in-degree at least $2^i$. All previous algorithms either required more time or more work. The minimum performance gain is a logarithmic factor improvement; on certain graph classes, accelerations by factors of more than $n^{0.4}$ can be achieved. The algorithm applies an approximate priority queue, which clusters nodes of similar degrees.

## AAR — The Algorithm Animation Repository

Rudolf Fleischer

Joint work with Pierluigi Crescenzi (Univ. of Florence), Nils Faltin (Univ. of Oldenburg), Chris Hundhausen (Univ. of Hawaii), Stefan Näher (Univ. of Trier), Guido Rössling (Univ. of Darmstadt), John Stasko (Georgia Institute of Technology), and Erkki Sutinen (Univ. of Joensuu.).

We introduce the Algorithm Animation Repository (AAR), a refereed collection of algorithm animations, animation tools, and other animation related material that is currently developed at the HKUST. The goal of the AAR is to provide a platform where good algorithm animations can be published and thus be made accessible to a wider audience, in particular for teaching purposes. The AAR was founded at last year's Dagstuhl Seminar on Software Visualization, with the above mentioned co-authors serving as the founding Editorial Board, under the chairmanship of R. Fleischer.

## Entropy-Compressed Indexes for Sequences

Jeff Vitter

Joint work with Roberto Grossi and Ankur Gupta.

We present results about the first index developed for string matching that provably achieves $o(m)$ search time and $o(n)$ words of storage in the worst case, where $m$ is the number of symbols in the search pattern, $n$ is the number of symbols in the sequence being searched, and the symbols are from an alphabet $\Sigma$. Based upon new compressed representations of two data structures for suffix arrays and suffix trees, our index uses $\epsilon^{-1} n \log|\Sigma| + O(n)$ bits to index a sequence, for any fixed constant $0 < \epsilon \leq 1$. For reasonable length patterns, the search time is $O(m \log|\Sigma| / \log n) = o(m)$, which is the length of the pattern in terms of computer words and is thus optimal. If the sequence is compressible, which is typically the case, we show how to get a corresponding reduction in the size of the index for the sequence, thus allowing significant space reductions in practice. More formally, we can replace the $\log|\Sigma|$ term in the space bounds by a high-order empirical entropy of the sequence.

## Dynamic Planar Convex Hull

Riko Jacob

Joint work with Gerth Brodal.

We determine the computational complexity of the dynamic convex hull problem in the planar case. We present a data structure that maintains a finite set of points in the plane under insertion and deletion of points in amortized $O(\log n)$ time. Here $n$ denotes the number of points in the set. The data structure supports the reporting of the extreme point of the set in some direction in worst-case $O(\log n)$ time. The space usage of the data structure is $O(n)$. We give a lower bound on the amortized asymptotic time complexity that matches the performance of our data structure.

## Real Time Traffic over the Internet and Per-Hop Behaviour

Andrej Brodnik

This is a joint work with Ulf Bodin, Johan Karlsson, Andreas Nilsson, and Olov Schelén (Luleå University of Technology, Lulea, Sweden).

Real Time (RT) traffic is becoming more and more important part of the Internet traffic. It consists of traffic serving applications such as video-conferencing, IP telephony, movie on demand, etc. In this contribution we study so called per-hop behaviour and management of IP traffic which includes also support for the RT traffic.
In particular, we present a data structure BinSeT (binary segment tree) – a combination of balanced tree and segment tree – that is used in link (hop) bandwidth (any resource, in general) reservation. The bandwidth reservation is one of the problems that needs to be solved to enable RT traffic delivery in time.
Besides, we present a system of FIFO buffers that permits an enhanced support for the RT traffic. The enhancement comes in the amount of transported data. In this scheme we have three kinds of a traffic: conforming – the one which is sent within the reserved bandwidth, excess – the one which is also part of the RT traffic and it is meant to improve the quality of transmission (if possible) and background – non-RT traffic sent by other applications. The later two are treated as best-effort traffic and compete for the same bandwidth.

# Thursday, February 28

## Cache Oblivious Divide-and-Conquer Algorithms Based on Merging

Rolf Fagerberg

Joint work with Gerth Stølting Brodal.

We adapt the distribution sweeping method to the cache oblivious model. Distribution sweeping is the name used for a general approach for divide-and-conquer algorithms where the combination of solved subproblems can be viewed as a merging process of streams. We demonstrate through a series of algorithms for specific problems the feasibility of the method in a cache oblivious setting. The problems all come from computational geometry, and are: orthogonal line segment intersection reporting, the all nearest neighbors problem, the 3D maxima problem, computing the measure of a set of axis-parallel rectangles, computing the visibility of a set of line segments from a point, batched orthogonal range queries, and reporting pairwise intersections of axis-parallel rectangles. These are the first non-trivial cache oblivious algorithms within computational geometry. Our basic building block is a simplified version of the cache oblivious sorting algorithm Funnelsort of Frigo et al., which may be of independent interest.

## Cache-Oblivious Priority Queue and Graph Algorithm Applications

Lars A. Arge

We present an optimal cache-oblivious priority queue data structure, supporting insertion, deletion, and deletemin operations in $O(\frac{1}{B}\log_{M/B}\frac{N}{B})$ amortized memory transfers, where $M$ and $B$ are the memory and block transfer sizes of any two consecutive levels of a multilevel memory hierarchy. In a cache-oblivious data structure, $M$ and $B$ are not used in the description of the structure. The bounds match the bounds of several previously developed external-memory (cache-aware) priority queue data structures, which all rely crucially on knowledge about $M$ and $B$. Priority queues are a critical component in many of the best known external-memory graph algorithms, and using our cache-oblivious priority queue we develop several cache-oblivious graph algorithms.

## Two Simplified Algorithms for Maintaining Order in a List

Michael A. Bender

Joint work with Erik D. Demaine and Martin Farach-Colton.

In the order-maintenance problem, the objective is to maintain a total order subject to insertions and deletions and precedence queries. This problem is an data structural workhorse in a variety of application areas ranging from programming languages to graph drawing to string algorithms. But the current solutions (notably Dietz and Sleator's optimal data structure) are complicated and consequently little understood and treated as blackboxes.
We present two simplified data structures for the order-maintenance problem. The first structure supports order queries in $O(1)$ worst-case time and insertions and deletions in $O(1)$ amortized time. The second deamortized structure supports all operations in $O(1)$ worst-case time. These optimal bounds match the bounds of two structures by Dietz and Sleator from 15 years ago, and our data structures and analyses are vastly simpler. As such, we make a major advance in the understanding of the order-maintenance problem. In particular, we transform the algorithms and their analyses from obscure black boxes into structure easily understood in a graduate algorithms class. We also present tighter analyses, settling an open problem by Dietz and Sleator, and validate experimentally that our analyses are nearly exact.

## Cache-Oblivious Traversal of a Dynamic List

Erik D. Demaine

This is joint work with Michael A. Bender, Richard Cole, and Martin Farach-Colton.

We consider the problem of designing a linked-list data structure with high data locality. Specifically, we design efficient cache-oblivious data structures that support three operations: (1) insert an element immediately after a specified element, (2) delete a specified element, and (3) traverse $k$ logically consecutive elements. When the memory block transfer size $B$ is known (cache aware), it is easy to obtain optimal bounds of constant time for an update and $O(\lceil k/B \rceil)$ time for a traversal. When $B$ is unknown (cache oblivious), we develop a data structure supporting $O((\log\log N)^{2+\epsilon}/B^{1-\epsilon})$ updates and nearly optimal traversals. We also consider several restricted forms of the problem.

## Optimal Solutions for the Temporal Precedence Problem

Athanasios Tsakalidis

Joint work with G. S. Brodal (BRICS (Basic Research in Computer Science), Department of Computer Science, University of Aarhus, Ny Munkegade, DK-8000 Arhus C, Denmark) and C. Makris, S. Sioutas, and K. Tsichlas (Department of Computer Engineering & Informatics, University of Patras, 26500 Patras, Greece and Computer Technology Institute (CTI) P.O.BOX 1192, 26110 Patras, Greece)

In this paper we refer to the *Temporal Precedence Problem* on *Pure Pointer Machines*. This problem asks for the design of a data structure, maintaining a set of stored elements and supporting the following two operations: *insert* and *precedes*. The operation $insert(a)$ introduces a new element $a$ in the structure, while the operation $precedes(a, b)$ returns true iff element $a$ was inserted before element $b$ temporally. In Ranjan et al. a solution was provided to the problem with worst-case time complexity $O(\log\log n)$ per operation and $O(n \log\log n)$ space, where $n$ is the number of elements inserted. It was also demonstrated that the *precedes* operation has a lower bound of $\Omega(\log\log n)$ for the *Pure Pointer Machine* model of computation. In this paper we present two simple solutions with linear space and worst-case constant insertion time. In addition, we describe two algorithms that can handle the $precedes(a, b)$ operation in $O(\log\log d)$ time, where $d$ is the temporal distance between the elements $a$ and $b$.
**Keywords:** Algorithms, Dynamic Data Structures, Computational Complexity.

## Designing Data Structures for the Average Case: Priority Queues and van Emde Boas Trees in Constant Time

J. Ian Munro

This is joint work with Michael Bender of SUNY Stony Brook and Erik Demaine of MIT.

We focus on the general issue of designing data structures to perform well in the average case. In particular, we introduce a priority queue supporting random insertions and a data structure supporting successor operations (union-split-find) with random updates. These data structures are explicitly tuned to perform in constant time not only on average but also with high probability on a random sequence of insertions and deletions. Furthermore, this behavior is maintained despite worst-case delete-min operations and worst-case successor queries. This work stands in contrast to most work on average-case data structures in that the innovation is in the algorithm design rather than the analysis.

## Decoupling: A Free(?) Spatial Lunch

Hanan Samet

A pair of new hierarchical spatial data structures that are based on decoupling of the space decomposition and the directory are reviewed and the ramifications of their deployment are presented. The first is the BV-tree of Freeston [1] which decouples the hierarchy inherent to the tree structure of a directory from the containment hierarchy associated with the process of recursive decomposition of the underlying space from which the data is drawn. This enables point queries to be executed by descending the same number of levels in the decomposition hierarchy even though the blocks in the directory hierarchy are not disjoint. The second is the PK tree of Wang, Yang, and Muntz [2] which decouples the tree structure of the partition process of the underlying space from that of the node hierarchy (i.e., the grouping process of the nodes resulting from the partition process) that makes up the directory. Instead of decomposing whenever there are more than $k$ objects in a block as in bucketing methods, it groups objects whenever there are fewer than $k$ objects in a block (termed instantiation). Applying a PK tree to a k-d tree (or any binary tree variant) results in nodes that have between $k$ and $2k-1$ objects which is analogous to a B-tree. The shape of the block that corresponds to the PK tree node is congruent to the shape of the blocks that result from the decomposition process that is used. For $N$ objects, the maximum depth is $O(N)$ instead of being dependent on the minimum separation between any two existing points which is the case for methods based on a regular decomposition of the underlying space.

1. M. Freeston. A general solution of the n-dimensional B-tree problem. In *Proceedings of the ACM SIGMOD Conference*, pages 80–91, San Jose, CA, May 1995.

2. W. Wang, J. Yang, and R. Muntz. PK-tree: a spatial index structure for high dimensional point data. In *Proceedings of the Fifth International Conference on Foundations of Data Organization and Algorithms (FODO)*, K. Tanaka and S. Ghandeharizadeh, eds., pages 27–36, Kobe, Japan, November 1998.

## Permutation Generation Methods

Robert Sedgewick

Generating all permutations of a set of objects is often useful in studying the structure of combinatorial problems. Even though we can certainly only consider small problem instances, such instances can provide insights needed to analyze larger instances.

Numerous methods for permutation generation have been proposed, some as long ago as the 17th century. One of the fastest, due to Heap, can be implemented such that, with appropriate optimizations, the cost per permutation is about the cost of two "store to memory" instructions. On a modern laptop, this method leads to permutation generation at a rate approaching billions of permutations per second.
For example, the performance of graph algorithms is dependent on the order in which edges are processed when the graph representation is constructed. A first step in understanding this dependence is to try all edge permutations for specific graphs and specific graph algorithms of interest.

## Faster Knapsack Approximations

Martin Fürer

The standard Knapsack Maximization Problem (select $I \subseteq \{1, \ldots, n\}$ maximizing $\sum_{i \in I} v_i$ subject to $\sum_{i \in I} s_i \leq b$) can be approximated by a factor of $1 - \frac{1}{k}$ in time $O((n + k^3) \log k)$ (previously $O(n \log k + k^4)$). For the Subset-Sum Problem (special case of knapsack with $s_i = v_i$) this approximation is achieved in time $O(n + k^3)$. The improvement is obtained by handling the insertion of a whole slot of items simultaneously. A slot consists of items of approximately the same value.

## Implicit Representation of B-Trees

Roberto Grossi

Joint work with G. Franceschini, I. Munro, L. Pagli.

In this talk we describe an implicit representation of B-trees suitable for external memory. Saving space is an important issue for large data sets. In the implicit representation, no auxiliary space is needed for the indexes (B-trees), just a suitable permutation of the $n$ keys in a contiguous area encodes the entire B-tree. This is also useful to transmit and dump only the permuted keys, since the index is encoded by the sequence of keys. A simple example of this idea is the classical heap, in which the $i$th key has its parent in position $\lfloor i/2 \rfloor$ (if $i > 1$) and its children (if any) in positions $2i$ and $2i + 1$, respectively. We want to support fast searching (under the comparison model) and fast updating of the permutation, as in standard B-trees. We describe how to store the $n$ keys and their B-tree in

$\lceil n/B \rceil$ disk pages, where $B$ is the size of the I/O block transfer. Our representation builds up on previous work by Munro, and supports searching and updating in $o(\log_B^2 n)$ I/Os in the general case, reducing to $O(\log_B n)$ I/Os when $B > \log n$ (a realistic assumption). The implementation can be used in main memory to get $O(\log^2 n / \log\log n)$ time for the operations.

## Two Bit-Probe Search

Venkatesh Raman

Joint work with Jaikumar Radhakrishnan and S. Srinivasa Rao.

We consider the classical dictionary problem of representing a set $S \subseteq \{0, 1, ...m-1\}$ such that $|S| \leq n$ to support membership queries in the bitprobe model. Here the space $s$ is counted in terms of the number of bits in the representation, and the time $t$ is counted as the number of bits probed to the representation.
We give an $O(m^{2/3})$ algorithm for representing two element sets using two probes. This is generalized to give a structure using $O(m^{1-1/(\lg n+2)})$ bits and $\lg\lg n+2$ bit probes. We also show that our two probe scheme for two element sets is optimal among a class of two probe schemes.

# Friday, March 1

## On the Performance of Randomized Search Heuristics on Simple Combinatorial Problems

Ingo Wegener

Joint work with Oliver Giel, Jens Scharnow, and Karsten Tinnefeld

Randomized search heuristics are an alternative if no problem-specific algorithm is available or in black-box optimization where the function to be optimized is not known and can be sampled only. Our aim is to understand the performance of randomized search heuristics on selected problem classes. Here well-studied problems like sorting (the maximization of the sortedness), shortest paths, and maximal matching are studied. Different measures of sortedness lead to optimization problems of different degrees of difficulty. A simple evolutionary algorithm can solve most of them in expected polynomial time but the measure counting the number of runs leads to an exponential-time algorithm. Simple-source-shortest-paths problems are easy only if considered as problems of multi-objective optimization. Many instances of the maximal matching problem can be solved in expected polynomial time – even without using the idea of augmenting paths. However, some instances turn out to be difficult for randomized search heuristics.

## Call Control in Rings

Christoph Ambühl

Joint work with Udo Adamy, Sai Anand, and Thomas Erlebach.

The call control problem is an important optimization problem encountered in the design and operation of communication networks. The goal of the call control problem in ring networks is to compute, for a given ring network with edge capacities and a set of paths in the ring, a maximum cardinality subset of the paths such that no edge capacity is violated. We give a polynomial-time algorithm to solve the problem optimally. The algorithm is based on a decision procedure that checks whether a solution with at least $k$ paths exists, which is in turn implemented by an iterative greedy approach operating in rounds. We show that the algorithm can be implemented efficiently and, as a by-product, we obtain a linear-time algorithm to solve the call control problem in chain networks optimally.

## Time Responsive External Data Structures for Moving Points

Jan Vahrenhold

This is joint work with Pankaj K. Agarwal and Lars Arge (Duke University).

We develop external data structures for storing points in one or two dimensions, each moving along a linear trajectory, so that a range query at a given time $t_q$ can be answered efficiently. The novel feature of our data structures is that the number of I/Os required to answer a query depends not only on the size of the data set and on the number of points in the answer but also on the difference between $t_q$ and the current time; queries close to the current time are answered fast, while queries that are far away in the future or in the past may take more time.

## On Max-Clique Trees with Applications

Prosenjit Bose

We design a data structure $T$ that maintains in $O(\log n)$ time per operation, a function $f : \mathbb{R} \to \mathbb{R}$ under the following query and update operations where $[a, b)$ is a continuous interval in $\mathbb{R}$.

1. Insert$(T, a, b, \delta)$: Increase the value of $f(x)$ by $\delta$ for all $x \in [a, b)$.
2. Delete$(T, a, b, \delta)$: Decrease the value of $f(x)$ by $\delta$ for all $x \in [a, b)$.
3. Max-Cover(): Return $\max\{f(x) : x \in \mathbb{R}\}$.
4. Max-Cover-Witness(): Return a value $x^*$ such that $f(x^*) = \max\{f(x) : x \in \mathbb{R}\}$.
5. Max-In$(a, b)$: Returns $\max\{f(x) : x \in [a, b)\}$.
6. Max-Witness-In$(a, b)$: Returns a value $x^*$ such that $f(x^*) = \max\{f(x) : x \in [a, b)\}$.

as well as the Min counter-parts of these queries. We present several applications such as how to translate a grid of size m so that the number of $k$-occupied cells is maximized or minimized in $O(knm \log^{O(1)}(nm))$ time.

## Optimal Finger Search Trees in the Pointer Machine

Gerth Stølting Brodal

Joint work with George Lagogiannis, Christos Makris, Athanasios Tsakalidis, and Kostas Tsichlas.

We develop a new finger search tree with worst-case constant update time in the Pointer Machine (PM) model of computation. This was a major problem in the field of Data Structures and was tantalizingly open for over twenty years while many attempts by researchers were made to solve it. The result comes as a consequence of the innovative mechanism that guides the balancing operations combined with incremental multiple splitting and fusion techniques over nodes.

## 2 List of Participants – Dagstuhl Seminar 02091

Date: 24.02.2002 – 01.03.2002
Title: Data Structures
webpage: `http://www.dagstuhl.de/02091/`


**Pankaj Kumar Agarwal**
Duke University
Dept. of Computer Science
LSRC Bldg.
Box 90129
NC 27708-0129 Durham
USA
phone: +1-919-660-6540
fax: +1-919-660-6519
e-mail: `pankaj@cs.duke.edu`
webpage: `http://www.cs.duke.edu/~pankaj`

**Susanne Albers**
Universität Freiburg
Institut für Informatik
Georges-Khler-Allee 79
D-79110 Freiburg
D
phone: +49-761-203-8040
fax: +49-761-203-8042
e-mail: `salbers@informatik.uni-freiburg.de`
webpage: `http://www.informatik.uni-freiburg.de/~salbers/`

**Stephen Alstrup**
The IT University of Copenhagen
Glentevej 67
DK-2400 Copenhagen
DK
phone: +45-38 16 88 22
fax: +45-38 16 88 99
e-mail: `stephen@itu.dk`
webpage: `http://www.itu.dk/people/stephen/`

**Christoph Ambuehl**
ETH Zürich
Institut für Theoretische Informatik
IFW 44
Clausiusstrasse 49
ETH-Zentrum
CH-8092 Zürich
CH
e-mail: `ambuehl@inf.ethz.ch`
webpage: `http://www.inf.ethz.ch/personal/ambuehl/`

**Lars Arge**
Duke University

Dept. of Computer Science
D205 LSRC
Box 90129
NC 27708-0129 Durham
USA
phone: +1-919-660-6557
fax: +1-919-660-6519
e-mail: large@cs.duke.edu
webpage: http://www.cs.duke.edu/~large/

**Michael Bender**
SUNY at Stony Brook
Dept. of Computer Science
NY 11794-4400 Stony Brook
USA
phone: +1-631-632-7835
fax: +1-631-632-8334
e-mail: bender@cs.sunysb.edu
webpage: http://www.cs.sunysb.edu/~bender

**Norbert Blum**
Universität Bonn
Institut für Informatik
Römerstr. 164
D-53117 Bonn
D
phone: +228-734-250
fax: +228-734-440
e-mail: blum@cs.uni-bonn.de
webpage: http://theory.cs.uni-bonn.de/blum

**Prosenjit Bose**
Carleton University
School of Computer Science
1125 Colonel By Drive
ON-K1S 5B6 Ottawa
CDN
phone: +1-613-520-43 33
e-mail: jit@scs.carleton.ca
webpage: http://www.scs.carleton.ca/~jit/

**Gerth Stølting Brodal**
University of Aarhus
BRICS - Dept. of Computer Science
Ny Munkegade
BG 540
DK-8000 Aarhus
DK
phone: +45-89 42 34 72
fax: +45-89 42 32 55
e-mail: gerth@brics.dk
webpage: http://www.brics.dk/~gerth/

**Andrej Brodnik**
University of Ljubljana
Institut of Mathematics, Physics & Mechanics
Jadranska 19
SI-1000 Ljubljana
SLO
(Also Luleå University of Technology, Lulea, Sweden)
phone: +386-1-476-6685
fax: +386-1-476-6500
e-mail: `brodnik@lage.cdt.luth.se`
webpage: `http://www.sm.luth.se/~brodnik/`

**Adam Buchsbaum**
AT&T Labs Research
Shannon Lab.
180 Park Avenue
P.O. Box 971
NJ 07932-0971 Florham Park
USA
phone: +1-973-360-8674
fax: +1-973-360-8077
e-mail: `alb@research.att.com`
webpage: `http://www.research.att.com/info/alb`

**Fabrizio d'Amore**
Università di Roma "La Sapienza"
Dipartimento di Informatica e Sistemistica
Via Salaria 113
I-00198 Roma
I
phone: +39-06-4991-8333
fax: +39-06-8530-0849
e-mail: `damore@dis.uniroma1.it`
webpage: `http://www.dis.uniroma1.it/~damore/`

**Frank Dehne**
Carleton University
School of Computer Science
Herzberg Labs.
1125 Colonel By Drive
ON-K1S 5B6 Ottawa
CDN
phone: +1-613-520-43 33
fax: +1-613-520-43 34
e-mail: `frank@dehne.net`
webpage: `http://www.dehne.net/`

**Erik Demaine**
MIT
Laboratory for Computer Science
200 Technology Square
MA 02139 Cambridge
USA

phone: +1-617-253-6871
fax: +1-617-253-0415
e-mail: edemaine@mit.edu
webpage: http://db.uwaterloo.ca/~eddemain/

**Martin Dietzfelbinger**
Technische Universität Ilmenau
Fakultät für Informatik und Automatisierung
Postfach 100565
D-98684 Ilmenau
D
phone: +49-3677-69-2656
fax: +49-3677-69-1237
e-mail: martin.dietzfelbinger@tu-ilmenau.de
webpage: http://eiche.theoinf.tu-ilmenau.de/person/dietzd.html

**Rolf Fagerberg**
Aarhus University
Dept. of Computer Science
BRICS
Ny Munkegade
DK-8000 Aarhus
DK
fax: +45-89 42 32 55
e-mail: rolf@brics.dk
webpage: http://www.daimi.au.dk/~rolf/

**Rudolf Fleischer**
Hong Kong University of Science & Technology
Dept. of Computer Science
Clear Water Bay, Kowloon
Hong Kong
HK
phone: +852-2358-8770
fax: +852-2358-1477
e-mail: rudolf@cs.ust.hk
webpage: http://www.cs.ust.hk/~rudolf/

**Paolo Franciosa**
Università di Roma "La Sapienza"
Dip. di Statistica, Probabilita e Statistiche Applicate
Piazzale Aldo Moro 5
I-00185 Roma
I
phone: +39-06-499-10496
fax: +39-06-495-9241
e-mail: paolo.franciosa@uniroma1.it
webpage: http://www.dis.uniroma1.it/~pgf/newhome.html

**Martin Fürer**
The Pennsylvania State University
Dept. of Computer Science and Engineering
328 Pond Laboratory

PA 16802 University Park
USA
(Also ETH Zürich, Switzerland)
phone: +1-814-863-1857
fax: +1-814-865-3176
e-mail: furer@cse.psu.edu, martin.fuerer@math.ethz.ch
webpage: http://www.cse.psu.edu/~furer/

**Roberto Grossi**
Università degli Studi di Pisa
Dipartimento di Informatica
Room O-II-11
Corso Italia 40
I-56125 Pisa
I
phone: +38-050-2212-793
fax: +39-050-2212-726
e-mail: grossi@di.unipi.it
webpage: http://www.di.unipi.it/~grossi/

**Torben Hagerup**
Universität Frankfurt
Institut für Informatik
Robert-Mayer-Str. 11-15
PF 11 19 32
D-60054 Frankfurt
D
phone: +49-697-982-8155
e-mail: hagerup@ka.informatik.uni-frankfurt.de
webpage: http://www.informatik.uni-frankfurt.de/~hagerup/

**Klaus Hinrichs**
Westfälische Wilhelms-Universität Münster
FB 10 Mathematik und Informatik
Einsteinstr. 62
D-48149 Münster
D
phone: +49-251-833-3752
fax: +49-251-833-3755
e-mail: hinrichs@math.uni-muenster.de
webpage: http://www.uni-muenster.de/math/inst/info/u/khh/

**Gisli R. Hjaltason**
RightOrder Inc.
3850 North First Street
CA 95134 San Jose
USA
fax: +1-408-321-6268
e-mail: gisli@mac.com
webpage: http://www.cs.umd.edu/~grh

**Riko Jacob**
University of Aarhus

BRICS - Dept. of Computer Science
Ny Munkegade
BG 540
DK-8000 Aarhus
DK
phone: +45-8942-3235
fax: +45-8942-3255
e-mail: rjacob@brics.dk
webpage: http://www.brics.dk/~rjacob

**Klaus Jansen**
Universität Kiel
Institut für Informatik und Prakt. Mathematik
Olshausenstr. 40
D-24098 Kiel
D
phone: +49-431-880-7501
fax: +49-431-880-7614
e-mail: kj@informatik.uni-kiel.de
webpage: http://www.informatik.uni-kiel.de/inf/Jansen/

**Haim Kaplan**
Tel Aviv University
School of Mathematical Sciences
Ramat-Aviv
69978 Tel-Aviv
IL
e-mail: haimk@math.tau.ac.il
webpage: http://www.math.tau.ac.il/~haimk/

**Rolf Klein**
Universität Bonn
Institut für Informatik
Institut I
Römerstr. 164
D-53117 Bonn
D
phone: +49-228-73-43 33
fax: +49-228-73-43 21
e-mail: rolf.klein@informatik.uni-bonn.de
webpage: http://tinos.informatik.uni-bonn.de/Mitarbeiter/klein.htm

**Manfred Kunde**
Technische Universität Ilmenau
Fakultät für Informatik und Automatisierung
FG Automaten und Formale Sprachen
Postfach 100565
D-98684 Ilmenau
D
phone: +49-3677-692-766
fax: +49-3677-69 12 37
e-mail: kunde@theoinf.tu-ilmenau.de
webpage: http://eiche.theoinf.tu-ilmenau.de/person/kunde.html

**Kim Skak Larsen**
University of Southern Denmark
Dept. of Mathematics and Computer Science
Campusvej 55
DK-5230 Odense
DK
phone: +45-65-50-23-28
fax: +45-65-93-26-91
e-mail: kslarsen@imada.sdu.dk
webpage: http://www.imada.sdu.dk/~kslarsen/

**Ulrich Lauther**
Siemens AG
Corporate Technology
CT SE 6
Otto-Hahn-Ring 6
D-81739 München
D
phone: +49-89-636 48834
fax: +49-89-636-42284
e-mail: ulrich.lauther@mchp.siemens.de

**Alejandro Lopez-Ortiz**
University of Waterloo
Dept. of Computer Science
200 University Avenue West
ON-N2L 3G1 Waterloo
CDN
e-mail: alopez-o@uwaterloo.ca
webpage: http://www.cs.unb.ca/~alopez-o

**Christos Makris**
University of Patras
Dept. of Comp. Eng. and Informatics
GR-26500 Patras
GR
fax: +30-61-960-322
e-mail: makri@ceid.upatras.gr

**Madhav Marathe**
Los Alamos National Laboratory
Basic & Applied Simulation Science D-2, MS M997
NM 87545 Los Alamos
USA
fax: +1-505-665-7464
e-mail: madhav@lanl.gov
webpage: http://fano.ics.uci.edu/cites/Author/Madhav-V-Marathe.htm

**Kurt Mehlhorn**
MPI für Informatik
Stuhlsatzenhausweg 85
D-66123 Saarbrücken
D

phone: +49-681-9325 100
fax: +49-681-9325 199
e-mail: mehlhorn@mpi-sb.mpg.de
webpage: http://www.mpi-sb.mpg.de/~mehlhorn

**Ulrich Carsten Meyer**
MPI für Informatik
Stuhlsatzenhausweg 85
D-66123 Saarbrücken
D
phone: +49-681-9325-125
fax: +49-681-9325-199
e-mail: umeyer@mpi-sb.mpg.de
webpage: http://www.mpi-sb.mpg.de/~umeyer/

**Ian Munro**
University of Waterloo
Dept. of Computer Science
200 University Avenue West
ON-N2L 3G1 Waterloo
CDN
phone: +1-519-888-45 67 ext. 44 33
fax: +1-519-885-12 08
e-mail: imunro@uwaterloo.ca
webpage: http://algonquin.uwaterloo.ca/~imunro/

**Hartmut Noltemeier**
Universität Würzburg
Institut für Informatik
Informatik I
Am Hubland
D-97074 Würzburg
D
phone: +49-931-888 5054 (5055)
fax: +49-931-888-46 00
e-mail: noltemei@informatik.uni-wuerzburg.de
webpage: http://www-info1.informatik.uni-wuerzburg.de/

**Stefan Näher**
Universität Trier
FB IV - Informatik
Universitätsring 15
D-54286 Trier
D
phone: +49 651 201-3275
fax: +49 651 201-3856
e-mail: naeher@informatik.uni-trier.de
webpage: http://www.informatik.uni-trier.de/~naeher/

**Rasmus Pagh**
University of Aarhus
BRICS - Dept. of Computer Science
Ny Munkegade

BG 540
DK-8000 Aarhus
DK
fax: +45-89 42 35 55
e-mail: pagh@daimi.au.dk
webpage: http://www.brics.dk/~pagh/

**Guido Proietti**
Università degli Studi di l'Aquila
Dipartimento di Informatica
Via Vetoio Coppito
I-67010 L'Aquila
I
fax: +39-0862-433-180
e-mail: proietti@univaq.it
webpage: http://galileo.dm.univaq.it/~proietti/

**Rajeev Raman**
University of Leicester
Dept. of Mathematics & Computer Science
University RD
LE1 7RH Leicester
GB
fax: +44-116-252-3604
e-mail: r.raman@mcs.le.ac.uk

**Venkatesh Raman**
The Institute of Mathematical Sciences
C.I.T. Campus
600113 Chennai
IND
fax: +91-044-254-1586
e-mail: vraman@imsc.ernet.in
webpage: http://www.imsc.ernet.in/~vraman/

**S.Srinivasa Rao**
University of Leicester
Dept. of Mathematics & Computer Science
University RD
LE1 7RH Leicester
GB
fax: +44-116-252 3915
e-mail: rao@mcs.le.ac.uk
webpage: http://imsc.ernet.in/~ssrao/

**Theis Rauhe**
The IT University of Copenhagen
Glentevej 67
DK-2400 Copenhagen
DK
fax: +45-38 11 41 39
e-mail: theis@itu.dk
webpage: http://www.itu.dk/people/theis/

**S.S. Ravi**
SUNY at Albany
Dept. of Computer Science
1400 Washington Avenue
NY 12222 Albany
USA
phone: +1-518-442-4278
fax: +1-518-442-5638
e-mail: `ravi@cs.albany.edu`
webpage: `http://www.cs.albany.edu/profiles/ravi.htm`

**Hanan Samet**
Univ. of Maryland at College Park
Dept. of Computer Science
MD 20742 College Park
USA
phone: +1-301-405 1755
fax: +1-301 314 9115
e-mail: `hjs@umiacs.umd.edu`
webpage: `http://www.cs.umd.edu/~hjs/`

**Robert Sedgewick**
Princeton University
Dept. of Computer Science
35 Olden Street
NJ 08544 Princeton
USA
phone: +1-609-258-4345
fax: +1-609-258-1771
e-mail: `rs@cs.princeton.edu`
webpage: `http://www.CS.Princeton.EDU/~rs/`

**Raimund Seidel**
Universität des Saarlandes
FR Informatik
Im Stadtwald
D-66123 Saarbrücken
D
phone: +49-681-302-4513
fax: +49-681-302-5576
e-mail: `rseidel@cs.uni-sb.de`
webpage: `http://www-tcs.cs.uni-sb.de/`

**Michiel Smid**
Universität Magdeburg
Fakultät für Informatik
Inst. für Simulation und Graphik
Universitätsplatz 2
D-39106 Magdeburg
D
phone: +49-391-67-18557
fax: +49-391-67-11164
e-mail: `michiel@isg.cs.uni-magdeburg.de`
webpage: `http://isgwww.cs.uni-magdeburg.de/~michiel/index.html`

**David Taylor**
ETH Zürich
Institut für Theoretische Informatik
CLW C 3
Clausiusstrasse 49
ETH-Zentrum
CH-8092 Zürich
CH
fax: +41-1-632-1399
e-mail: taylor@inf.ethz.ch
webpage: http://www.inf.ethz.ch/~taylor/

**Athanasios Tsakalidis**
University of Patras
Dept. of Comp. Eng. and Informatics
GR-26500 Patras
GR
phone: +30-610-997 894
fax: +30-610-960322 + 960374
e-mail: tsak@cti.gr
webpage: http://www.tsakalidis.gr/

**Esko Ukkonen**
University of Helsinki
Dept. of Computer Science
Teollisuuskatu 23
P.O. Box 26
FIN-00014 Helsinki
FIN
phone: +358-9-708-4172
fax: +358-9-708-4441
e-mail: ukkonen@cs.Helsinki.FI
webpage: http://www.cs.helsinki.fi/u/ukkonen/

**Jan Vahrenhold**
Westfälische Wilhelms-Universität Münster
FB 10 Mathematik und Informatik
Einsteinstr. 62
D-48149 Münster
D
phone: +49-251-83-38444
fax: +49-251-83-33755
e-mail: jan@math.uni-muenster.de
webpage: http://wwwmath.uni-muenster.de/cs/u/jan/

**Jeffrey S. Vitter**
Duke University
Dept. of Computer Science
D315 Levine Science Research Center
Box 90129
NC 27708-0129 Durham
USA
phone: +1-919-660-6503

fax: +1-919-660-6519
e-mail: jsv@cs.duke.edu
webpage: http://www.cs.duke.edu/~jsv/

**Dorothea Wagner**
Universität Konstanz
FB Informatik & Informationswissenschaft
Universitätsstr. 10
D 78
D-78457 Konstanz
D
phone: +49-7531-88-2893
fax: +49-7531-88-3577
e-mail: Dorothea.Wagner@uni-konstanz.de
webpage: http://www.fmi.uni-konstanz.de/~wagner/

**Ingo Wegener**
Universität Dortmund
FB Informatik II
D-44221 Dortmund
D
phone: +49-231-755-2776
fax: +49-231-755-2047
e-mail: wegener@ls2.informatik.uni-dortmund.de
webpage: http://ls2-www.informatik.uni-dortmund.de/~wegener/

**Peter Widmayer**
ETH Zürich
Institut für Theoretische Informatik
CLW
Clausiusstrasse 49
ETH-Zentrum
CH-8092 Zürich
CH
phone: +41-1-632-74 00
fax: +41-1-632-1399
e-mail: widmayer@inf.ethz.ch
webpage: http://www.inf.ethz.ch/personal/widmayer/

**Christian Worm Mortensen**
The IT University of Copenhagen
IT-C
Glentevej 67
DK-2400 Copenhagen
DK
fax: +45-38 16 88 99
e-mail: cworm@it-c.dk