10091 Abstracts Collection
Data Structures
— Dagstuhl Seminar —

Lars Arge[1], Erik Demaine[2] and Raimund Seidel[3]

[1] Univ. of Aarhus, DK
large@madalgo.au.dk
[2] MIT - Cambridge, US
edemaine@mit.edu
[3] Saarland University, DE
rseidel@cs.uni-saarland.de

Abstract. From February 28th to March 5th 2010, the Dagstuhl Seminar 10091 "Data Structures" was held in Schloss Dagstuhl – Leibniz Center for Informatics. It brought together 45 international researchers to discuss recent developments concerning data structures in terms of research, but also in terms of new technologies that impact how data can be stored, updated, and retrieved. During the seminar a fair number of participants presented their current research and open problems where discussed. This document first briefly describes the seminar topics and then gives the abstracts of the presentations given during the seminar.

Keywords. Data structures, information retrieval, complexity, algorithms

10091 Summary - Data Structures

The purpose of this workshop was to discuss recent developments in various aspects of data structure research, and also to familiarize the community with some of the problems that arise in the context of modern commodity parallel hardware architectures, such as multicore and GPU architectures. Thus while several attendees reported on progress on (twists on) old fundamental problems in data structures — e.g. Gerth Brodal, Rolf Fagerberg, John Iacono and Siddharrha Sen on search tree and dictionary structures, Bob Tarjan on heaps, Kasper D. Larsen and Peyman Afshani on range search data structures, and Peter Sanders and Michiel Smid on proximity data structures — there were also very inspiring presentations on new models of computation by Erik Demaine and on data structures on the GPU by John Owens. The latter presentation was one of the highlights of the seminar, and provided the attendees a good overview over possibilities and challenges in connection with design of data structures for GPU hardware. The seminar was attended by 45 international researchers, resulting in a congenial and productive atmosphere, which resulted in countless discussions and collaborations. The Dagstuhl atmosphere provided just the right environment for all of this.

Dagstuhl Seminar Proceedings 10091
Data Structures
http://drops.dagstuhl.de/opus/volltexte/2010/2686

Fractal Tree Databases

Michael A. Bender and Martin Farach-Colton

Insertion bottlenecks lie at the heart of database and file-system innovations, best practices, and system workarounds. Most databases and file systems are based on B-tree data structures, and suffer from the performance cliffs and unpredictable run times of B-trees. In this talk, I introduce the Fractal Tree data structure and explain how it provides dramatically improved performance in both theory and in practice.

From a theoretical perspective, if B is the block-transfer size, the B-tree performs $O(\log_B N)$ block transfers per insert in the worst case. In contrast, the Fractal Tree structure performs $O((\log_B N)/B)$ memory transfers per insert, which translates to run-time improvements of two orders of magnitude.

To relate that theory to practice, I present an algorithmic model for B-tree performance bottlenecks. I explain how the bottlenecks affect best practice and how database designers typically modify B-trees to try to mitigate the bottlenecks. Then I show how Fractal Tree structures can attain faster insertion rates, intuitively by transforming disk-seek bottlenecks into disk-bandwidth bottlenecks

I conclude with performance results. Tokutek has developed a transaction-safe Fractal-Tree storage engine for MySQL. I present performance results, showing how the system can maintain rich indexes more efficiently than B-trees. Surprisingly, Fractal Tree structures seem to maintain their order-of-magnitude competitive advantage over B-trees on both traditional rotating media as well as SSDs.

Keywords: Fractal Tree, Streaming B-tree, Database, Tokutek

Joint work of: Bender, Michael A.; Farach-Colton, Martin

On List Update with Locality of Reference

Susanne Albers (HU Berlin, DE)

We present a comprehensive study of the list update problem with locality of reference.

More specifically, we present a combined theoretical and experimental study in which the theoretically proven and experimentally observed performance guarantees of algorithms match or nearly match.

In the first part of the paper we introduce a new model of locality of reference that is based on the natural concept of runs. Using this model we develop refined theoretical analyses of popular list update algorithms. The second part of the paper is devoted to an extensive experimental study in which we have tested the algorithms on traces from benchmark libraries. It shows that the theoretical and experimental bounds differ by just a few percent.

Our new bounds are substantially lower than those provided by standard competitive analysis.

Another result is that the elegant Move-To-Front strategy exhibits the best performance, which confirms that it is the method of choice in practice.

Keywords: Self-organizing list; competitive analysis; experimental study

Joint work of: Albers, Susanne; Lauer, Sonja

See also: Proc. 35th International Colloquium on Automata, Languages and Programming, ICALP 2008

Cache-Oblivious Dynamic Dictionaries with Optimal Update/Query Tradeoff

Gerth Stolting Brodal (Aarhus University, DK)

Several existing cache-oblivious dynamic dictionaries achieve $O(\log_B N)$ memory transfers per operation (or slightly better $O(\log_B \frac{N}{M})$ transfers), where N is the number of items stored, M is the memory size, and B is the block size, which matches the classic B-tree data structure. One recent structure achieves the same query bound and a sometimes-better amortized update bound of $O\left(\frac{1}{B^{\Theta(1/(\log\log B)^2)}}\log_B N + \frac{1}{B}\log^2 N\right)$ memory transfers. This paper presents a new data structure, the *xDict*, implementing predecessor queries in $O(\frac{1}{\epsilon}\log_B \frac{N}{M})$ worst-case memory transfers and insertions and deletions in $O\left(\frac{1}{\epsilon B^{1-\epsilon}}\log_B \frac{N}{M}\right)$ amortized memory transfers, for any constant ϵ with $0 < \epsilon < 1$. For example, the xDict achieves subconstant amortized update cost when $N = M\,B^{o(B^{1-\epsilon})}$, whereas the B-tree's $\Theta(\log_B \frac{N}{M})$ is subconstant only when $N = o(MB)$, and the previously obtained $\Theta\left(\frac{1}{B^{\Theta(1/(\log\log B)^2)}}\log_B N + \frac{1}{B}\log^2 N\right)$ is subconstant only when $N = o(2^{\sqrt{B}})$.

The xDict attains the optimal tradeoff between insertions and queries, even in the broader external-memory model, for the range where inserts cost between $\Omega(\frac{1}{B}\lg^{1+\epsilon} N)$ and $O(1/\lg^3 N)$ memory transfers.

Keywords: Cache-oblivious, dictionary

Joint work of: Brodal, Gerth Stølting; Demaine, Erik D.; Fineman, Jeremy T.; Iacono, John; Langerman, Stefan; Munro, J. Ian

Full Paper:
http://www.siam.org/proceedings/soda/2010/SODA10_117_brodalg.pdf

See also: In Proc. 21st Annual ACM-SIAM Symposium on Discrete Algorithms, pages 1448-1456, 2010

Hard problems, meta-heuristics and parallelisation

Andrej Brodnik (University of Primorska, SI)

Public transportation optimisation problem can be split into a number of sub-problems, each of which is hard on its own – some are even NP-hard. To solve these (sub-)problems we need to employ various meta-heuristics. The particular problem we are solving is MDVSP (Multi-Depot Vehicle Scheduling Problem) which we transform into an IP (Integer Programming) problem. The later is solved using Linear programming and Branch&Bound meta-heuristic.

Secondly, we employ parallelisation to further speed-up and improve the quality of solution. We investigated the influence of combination of parallelisation and meta-heuristics on solving the maximum independent set problem (MIS). We got a reasonable quality improvement and speed-up, however for significant improvements we seem to need to improve the algorithms fundamentally. From the results seems that particular effort should be put into a diversification phase.

Keywords: Meta-heuristic, parallelisation

Joint work of: Brodnik, Andrej; Paš, David; Bèkèsi, Jòzsef; Krèsz, Miklòs; Cvahte, Rok

New Models of Computation

Erik D. Demaine (MIT - Cambridge, US)

Two models of computation deserve further study.

RALA (Reconfigurable Asynchronous Logic Automaton) is like digital circuits but where communication takes time proportional to distance. Equivalently, RALA is like a 2D or 3D grid of 1-bit stream processors connected to neighbors.

Reversible computing uses zero (or little) energy to compute in a way that can be undone, and costs energy for creating or destroying new bits.

What are optimal algorithms in these models?

Keywords: Parallel, geometry, energy, algorithms

Tight Thresholds for Cuckoo Hashing via XORSAT

Martin Dietzfelbinger (TU Ilmenau, DE)

We settle the question of tight thresholds for offline cuckoo hashing.

The problem can be stated as follows: we have n keys to be hashed into m buckets each capable of holding a single key.

Each key has $k \geq 3$ (distinct) associated buckets chosen uniformly at random and independently of the choices of other keys. A hash table can be constructed successfully if each key can be placed into one of its buckets. We seek thresholds α_k such that, as n goes to infinity, if $n/m \leq \alpha$ for some $\alpha < \alpha_k$ then a hash table can be constructed successfully with high probability, and if $n/m \geq \alpha$ for some $\alpha > \alpha_k$ a hash table cannot be constructed successfully with high probability. Here we are considering the offline version of the problem, where all keys and hash values are given, so the problem is equivalent to previous models of multiple-choice hashing. We find the thresholds for all values of $k > 2$ by showing that they are in fact the same as the previously known thresholds for the random k-XORSAT problem. We then extend these results to the setting where keys can have differing number of choices, and provide evidence in the form of an algorithm for a conjecture extending this result to cuckoo hash tables that store multiple keys in a bucket.

Keywords: Hashing, randomized, matching, random graphs, hypergraphs

Joint work of: Dietzfelbinger, Martin; Goertdt, Andreas; Mitzenmacher, Michael; Montanari, Andrea; Pagh, Rasmus; Rink, Michael

See also: Martin Dietzfelbinger, Andreas Goerdt, Michael Mitzenmacher, Andrea Montanari, Rasmus Pagh, Michael Rink: Tight Thresholds for Cuckoo Hashing via XORSAT CoRR abs/0912.0287: (2009)

Number Systems and Data Structures

Amr Elmasry (MPI für Informatik - Saarbrücken, DE)

The interrelationship between numerical representations and data structures is efficacious. However, in many write-ups such connection has not been made explicit. As far as we know, their usage was first discussed in the seminar notes by Clancy and Knuth. Early examples of data structures relying on number systems include finger search trees and binomial queues. In this talk, we survey some known number systems and their usage in existing worst-case efficient data structures. We formalize properties of number systems and requirements that should be imposed on a number system to guarantee efficient performance on the corresponding data structures. We introduce two new number systems: the strictly-regular system and the five-symbol skew system. We illustrate how to perform operations on the two number systems and give applications for their usage to implement worst-case efficient data structures. We also give a simple method that extends any number system supporting increments to support decrements using the same number of digit flips. The strictly-regular system [1] is a compact system that supports increments and decrements in constant number of digit flips. Compared to other number systems, the strictly-regular system has distinguishable properties. It is superior to the regular system for its efficient support to decrements, and superior to the extended-regular system for being

more compact by using three symbols instead of four. To demonstrate the applicability of the new number system, we modify Brodal's meldable priority queues making delete require at most 2 lg n+O(1) element comparisons (improving the bound from $7 \lg n + O(1)$) while maintaining the efficiency and the asymptotic time bounds for all operations. The five-symbol skew system [2] also supports increments with a constant number of digit flips. In this number system the weight of the ith digit is $2^i - 1$, and hence it can be used to implement efficient structures that rely on complete binary trees. As an application, we implement a priority queue as a forest of heap-ordered complete binary trees. The resulting data structure guarantees $O(1)$ worst-case cost per insert and $O(\lg n)$ worst-case cost per delete.

Keywords: Number systems, data structures, priority queues, worst-case performance, arithmetic operations
References:

1. A. Elmasry, C. Jensen and J. Katajainen, Strictly-regular number system and data struc- tures, 12th Scandinavian Symposium and Workshops on Algorithm Theory (2010), Bergen, Norway. To appear.
2. A. Elmasry, C. Jensen and J. Katajainen, The magic of a number system, 5th International Conference on Fun with Algorithms (2010), Ischia, Italy, in LNCS 6099, 156Ű165.

An O(log log n)-Competitive Binary Search Tree with O(log n) Worst-Case Search Time

Rolf Fagerberg (Univ. of Southern Denmark - Odense, DK)

We present the zipper tree, an $O(\log \log n)$-competitive online binary search tree that performs each access in $O(\log n)$ worst-case time. This shows that for binary search trees, optimal worst-case access time and near-optimal amortized access time can be guaranteed simultaneously.

Keywords: Binary search trees, competitive ratio, worst case access time

Joint work of: Bose, Prosenjit; Douïeb, Karim; Dujmović, Vida; Fagerberg, Rolf

Space lower bounds for succinct representation of graphs

Arash Farzan (MPI für Informatik - Saarbrücken, DE)

We consider the problem of encoding a graph with n vertices and m edges compactly supporting adjacency, neighborhood and degree queries in constant time in the log(n)-bit word RAM model.

The adjacency query asks whether there is an edge between two vertices, the neighborhood query reports the neighbors of a given vertex in constant time per neighbor, and the degree query reports the number of incident edges to a given vertex.

We study the problem in the context of succinctness, where the goal is to achieve the optimal space requirement as a function of n and m, to within lower order terms. We prove a lower bound in the cell probe model that it is impossible to achieve the information-theory lower bound within lower order terms unless the graph is too sparse (namely $m = o(n^\delta)$ for any constant $\delta > 0$) or too dense (namely $m = o(n^{2-\delta})$ for any constant $\delta > 0$).

Furthermore, we present a succinct encoding for graphs for all values of n, m supporting queries in constant time. The space requirement of the representation is always within a multiplicative $1 + \epsilon$ factor of the information-theory lower bound for any arbitrarily small constant $\epsilon > 0$. This is the best achievable space bound according to our lower bound where it applies. The space requirement of the representation achieves the information-theory lower bound tightly to within lower order terms when the graph is sparse ($m = o(n^\delta)$ for any constant $\delta > 0$).

Keywords: Succinct representations, Graph compression

Induced halpotyping

Rudolf Fleischer (Fudan University - Shanghai, CN)

Over the last few years, haplotype inference has become one of the central problems in algorithmic bioinformatics [1]. One of the two major approaches to haplotype inference is parsimony haplotyping: Given a set of genotypes, the task is to find a minimum-cardinality set of haplotypes that explains the input set of genotypes. A genotype can be seen as a length-m string over the alphabet $\{0, 1, 2\}$, while a haplotype can be seen as a length-m string over the alphabet $\{0, 1\}$. A set H of haplotypes explains, or resolves, a set G of genotypes if for every $g \in G$ there is either an $h \in H$ with $g = h$ (trivial case), or there are two haplotypes $h1$ and $h2$ in H such that, for all $i \in \{1, \ldots, m\}$, if g has letter 0 or 1 at position i, then both $h1$ and $h2$ have this letter at position i, and if g has letter 2 at position i, then one of $h1$ or $h2$ has letter 0 at position i while the other one has letter 1.

Parsimony haplotyping is NP-hard, and numerous algorithmic approaches based on heuristics and integer linear programming methods are applied in practice [1]. There is also a growing list of combinatorial approaches (with provable performance guarantees) including the identification of polynomial-time solvable special cases, approximation algorithms, and fixed-parameter algorithms [2]. We propose improved fixed-parameter tractability results (where the parameter is the size of the haplotype set) which also apply to the practically important constrained case, where we can only use haplotypes from a given set. Furthermore, we show that the problem becomes polynomial-time solvable if the given set of

genotypes is complete, i.e., contains all possible genotypes that can be explained by the set of haplotypes.

References:

1. D. Catanzaro and M. Labb. The pure parsimony haplotyping problem: Overview and computational advances. Int. Trans. in Operational Research, 16(5):561−584, 2009.
2. M. R. Fellows, T. Hartman, D. Hermelin, G. M. Landau, F. A. Rosamond, and L. Rozen- berg. Haplotype inference constrained by plausible haplotype data. In Proc. 20th CPM, volume 5577 of LNCS, pages 339−352. Springer, 2009.

Keywords: Fpt algorithm, parsimony haplotyping

Joint work of: Fleischer, Rudolf; Guo, Jiong; Niedermeier, Rolf; Uhlmann, Johannes; Wang, Yihui; Weller, Mathias; Wu, Xi

See also: To appear at CPM 2010

A strengthened analysis of an algorithm for Dominating Set in planar graphs

Torben Hagerup (Universität Augsburg, DE)

Alber et al. presented an algorithm for computing a dominating set of size at most k, if one exists, in an undirected planar n-vertex graph and bounded its execution time by $O(8^k n)$.

Here it is shown that the algorithm performs better than claimed by its authors.

More significantly, if $k \leq n/19$, even a much simplified version of the algorithm runs in $O(7^k n)$ time.

Keywords: Dominating Set, planar graphs, parameterization, FTP algorithms

Mergeable Dictionaries

John Iacono (Polytechnic Institute of NYU - Brooklyn, US)

A data structure is presented for the Mergeable Dictionary abstract data type, which supports the following operations on a collection of disjoint sets of totally ordered data: Predecessor- Search, Split and Union.

While Predecessor-Search and Split work in the normal way, the novel operation is Union. While in a typical mergeable dictionary (e.g. 2-4 Trees), the Union opera- tion can only be performed on sets that span disjoint intervals in keyspace, the structure here has no such limitation, and permits the merging of arbitrarily interleaved sets. Our data structure supports all operations, including Union, in $O(\log n)$ amortized time, thus showing that interleaved Union operations can be sup- ported at no additional cost vis-a-vis disjoint Union operations.

Keywords: Data structures, amortized analysis

Joint work of: Iacono, John; Özkan, Özgür

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2010/2685

A Distributed Polylogarithmic Time Algorithm for Self-Stabilizing Skip Graphs

Riko Jacob (TU München, DE)

In the setting of Peer-to-Peer networks, we like to have a topology that has low degree, low diameter and allows easy routing.

Additionally, this topology should be created in a local and distributed fashion from any initial state.

Here, we consider to use the Skip+ Graph and show that a natural topologically self stabilizing construction algorithm takes $O(\log^2 n)$ parallel rounds to converge to this graph.

Joint work of: Jacob, Riko; Richa, Andrea; Scheideler, Christian; Schmid, Stefan; Täubig, Hanjo

Optimal 3-d Orthogonal Range Reporting

Kasper Dalgaard Larsen (Aarhus University, DK)

Orthogonal range reporting is the problem of storing a set of n points in d-dimensional space, such that the k points in an axis-orthogonal query box can be reported efficiently. While the 2-d version of the problem was completely characterized in the pointer machine model more than two decades ago, this is not the case in higher dimensions.

In this talk we provide a space optimal pointer machine data structure for 3-d orthogonal range reporting that answers queries in $O(\log n + k)$ time. Thus we settle the complexity of the problem in 3-d. We use this result to obtain improved structures in higher dimensions, namely structures with a $\log n/\log\log n$ factor increase in space and query time per dimension. Thus for $d \geq 3$ we obtain a structure that both uses optimal $O(n(\log n/\log\log n)^{d-1})$ space and answers queries in the best known query bound $O(\log n(\log n/\log\log n)^{d-3} + k)$.

Keywords: Orthogonal range reporting, pointer machine

Joint work of: Afshani, Peyman; Arge, Lars; Larsen, Kasper Dalgaard

Local Strategies for Building Geometric Formations

Friedhelm Meyer auf der Heide (Universität Paderborn, DE)

Local Strategies for Building Geometric Formations Friedhelm Meyer auf der Heide Heinz Nixdorf Institute & Computer Science Department University of Paderborn

Consider a scenario with a set of autonomous mobile robots having initial positions in the plane.

Their goal is to move in such a way that they eventually reach a prescribed formation. Such a formation may be a straight line between two given endpoints (short communication chain), a circle or any other geometric pattern, or just one point (gathering problem). In this talk, I consider simple local strategies for such robotic formation problems: the robots are limited to see only robots within a bounded radius. Thus, their decisions where to move next are solely based on the relative positions of robots within the bounded radius.

I will survey several recent results on such formation problems, and will focus on a local algorithm that performs gathering in a quadratic number of rounds. All previous algorithms with a proven time bound assume global view on the positions of all robots.

Joint work of: Degener, Bastian; auf der Heide, Friedhelm Meyer; Kempkes, Barbara

I/O-Efficient Contour Queries on Terrains

Thomas Moelhave (Duke University, US)

Given a terrain in the form of a triangulation of the plane with a height function associated to vertices (and linearly interpolated within the edges and triangles), we investigate the problem of I/O-efficiently answering contour queries: Given a regular height value l and a triangle t of the terrain which intersects the level-set at height l as input, the output of the query is the list of the edges of the connected component of the level-set at height l that intersect t, reported in clockwise or counter-clockwise order. Notice that contour queries are different from level-set queries in that only one contour out of all those in the corresponding level set is required to be reported. We present an I/O-efficient data structure of linear size that can be used to answer a contour query in $O(\log_B N + T/B)$ I/Os, where T is the number of triangles in the contour. The data structure can be constructed using $O(\text{sort}(N))$ I/Os.

Keywords: External memory, I/O, algorithm, data structure, contour, gis, contour map

Joint work of: Agarwal, Pankaj K.; Mølhave, Thomas; Sadri, Bardia

Producing Partial Orders and Finishing the Sort

J. Ian Munro (University of Waterloo, CA)

We examine two problems related to (partial) sorting. The first is: given an arbitrary partial order and a set of elements with a total ordering, arrange the elements so that the partial order is respected ... using as few comparisons as possible. The second is, given the elements in a partial order, complete the sort. The second problem dates back to the mid 1970's and the result of Mike Fredman showing this can be done in $\log(E) + O(n)$ comparisons, where E denotes the number of linear extensions of the partial order, though the technique took exponential time to determine what comparisons should be performed. The author, and others, had considered the first problem shortly after this, though it was not until the 1989 that Andrew Yao showed it can be solved using the "order information theoretic" $O(\log(n!) - \log(E) + n)$ comparisons ... again requiring exponential time to determine what to do. In 1995 Kahn and Kim gave a polynomial time technique to solve this problem in the same order of comparisons, it was based on the ellipsoid method and the notion of graph entropy. We give simpler polynomial time methods to solve both problems in a number of comparisons differing from the information theoretic lower bounds by "lower order plus linear" terms; again the key notions and proofs come from graph entropy. For partial sorting we determine a multiple selection problem, that is a restriction of the desired partial order, and use the method of Kaligosi et al for multiple selection. To "complete the sort" we use a similar entropy based approach to find a set of chains that is "relaxation" of the partial order and then merge them.

This work appears in STOC 2009, and 2010. It feels great to knock off a problem that you first worked on over 30 years ago.

Keywords: Sorting, partial order, graph entropy

Joint work of: Cardinal, Jean; Fiorini, Sam; Joret, Gwenael; Jungers, Raph; Munro, J. Ian

How to Grow Your Balls

Mihai Patrascu (AT&T Research - Florham Park, US)

We give the first improvement to the space/approximation trade-off of distance oracles since the seminal result of Thorup and Zwick [STOC'01].

For unweighted graphs, our distance oracle has size $O(n^{5/3})$ and, when queried about vertices at distance d, returns a path of length $2d+1$.

For weighted graphs with $m = n^2/\alpha$ edges, our distance oracle has size $O(n^2/\alpha^{1/3})$ and returns a factor 2 approximation.

Based on a widely believed conjecture about the hardness of set intersection queries, we show that a 2-approximate distance oracle requires space $\tilde{\Omega}(n^2/\sqrt{\alpha})$. For unweighted graphs, this implies a $\tilde{\Omega}(n^{1.5})$ space lower bound to achieve approximation $2d+1$.

Keywords: Distance oracles, graphs, routing, shortest paths

Analyzing Data Structures with Forbidden 0-1 Matrices

Seth Pettie (Univ. of Michigan - Ann Arbor, US)

In this paper we improve, reprove, and simplify several theorems on the performance of data structures based on path compression and search trees. We apply a technique very familiar to computational geometers but still foreign to many researchers in (non-geometric) algorithms and data structures, namely, to bound the complexity of an object via its *forbidden substructures*.

To analyze an algorithm or data structure in the forbidden substructure framework one proceeds in three discrete steps. First, one *transcribes* the behavior of the algorithm as some combinatorial object M; for example, M may be a graph, sequence, permutation, matrix, set system, or tree.

(The size of M should ideally be linear in the running time.) Second, one shows that M excludes some forbidden substructure P, and third, one bounds the size of any object avoiding this substructure.

The power of this framework derives from the fact that M lies in a more pristine environment and that upper bounds on the size of a P-free object M may be reused in different contexts.

Among our results, we present the first asymptotically sharp bound on the length of arbitrary path compressions on arbitrary trees, improving analyses of Tarjan and Seidel and Sharir. We reprove the linear bound on postordered path compressions, due to Lucas and Loebel and Nesetril, the linear bound on deque-ordered path compressions, due to Buchsbaum, Sundar, and Tarjan, and the sequential access theorem for splay trees, originally due to Tarjan. We disprove a conjecture of Aronov et al. related to the efficiency of their data structure for half-plane proximity queries and provide a significantly cleaner analysis of their structure.

A Query Lower Bound for Orthogonal Range Reporting

Afshani Peyman (Aarhus University, DK)

Orthogonal range reporting is the problem of storing a set of points in d-dimensional space, such that the points in an axis-orthogonal query box can be reported efficiently.

This is a fundamental problem in several fields.

We prove the first non-trivial query lower bound for the problem in the pointer machine model of computation. We believe this is an important result that also leads to some intriguing open problems for future research.

Keywords: Computatinal geometry, range searching, orthogonal range reporting, lower bounds in pointer machine

Joint work of: Peyman, Afshani; Arge, Lars; Larsen, Kasper Dalgaard

Streaming Algorithms for extent problems in high dimensions

Sharathkumar Raghvendra (Duke University, US)

We develop (single-pass) streaming algorithms for maintaining extent measures of a stream S of n points in $\mathbb{R}^d$.

We focus on designing streaming algorithms whose working space is polynomial in d (poly(d)) and sub-linear in n. For the problems of computing diameter, width and minimum enclosing ball of S, we obtain lower bounds on the worst-case approximation ratio of any streaming algorithm that uses poly(d) space. On the positive side, we introduce the notion of blurred ball cover and use it for answering approximate farthest-point queries and maintaining approximate minimum enclosing ball and diameter of S.

We describe a streaming algorithm for maintaining a blurred ball cover whose working space is linear in d and independent of n.

Keywords: Streaming Algorithms, Computational Geometry

Optimal Trade-Off for Succinct String Indexes

Rajeev Raman (University of Leicester, GB)

Let s be a string of n characters drawn from an integer alphabet of size $\sigma \leq n$. Access to s is solely through access(i), a read-only operation that *probes* s and returns the character in position i of s.

We wish to support two queries on s: select(c, j), returning the position in s containing the j'th occurrence of c, and rank(c, p), counting how many occurrences of c are found in the first p positions of s.

The efficiency of a solution for the above problem is measured primarily in terms of the number of probes to s, and the number of additional bits required to store an *index* on s. An index is any, possibly precomputed, auxiliary information on s; the space used for the index is additional to any space required for the representation of s itself.

We give matching upper and lower bounds for this problem showing that, e.g. for $t \leq \log\sigma/\log\log\sigma$, any index requires $\Theta\left(\frac{n\log\sigma}{t}\right)$ bits on s to support the above operations using at most t probes. This improves the lower bounds given by Golynski [*Theor. Comput. Sci.* extbf387 (2007)] [PhD thesis] and the upper bounds of Barbay et al. [SODA'07].

We also present new results in the *non-systematic* model, where we count the total space for representing s together with any index. We describe a data structure that encodes the string s in $nH_k(s) + O(n\log\sigma/\log\log\sigma)$ bits in order to support rank(c, p) and select(c, j) in $O(\log\log\sigma)$ time and access(i) in $O(1)$ time. We thus improve on Barbay et al. [SODA'07] and match a lower bound of Golynski [SODA'09] for large alphabets.

Keywords: Data Structures, Succinct Data Structures, Lower Bounds, Rank and Select operations, Monotone Hashing

Joint work of: Grossi, Roberto; Orlandi, Alessio; Raman, Rajeev

Simple and Fast Nearest Neighbor Search

Peter Sanders (KIT - Karlsruhe Institute of Technology, DE)

We present a simple randomized data structure for two-dimensional point sets that allows fast nearest neighbor queries in many cases. An implementation outperforms several previous implementations for commonly used benchmarks.

Keywords: Computation geometry, nearest neighbor, algorithm engineering, randomized incremental construction

Joint work of: Birn, Marcel; Holtgrewe, Manuel; Sanders, Peter; Singler, Johannes

Full Paper:
http://www.siam.org/proceedings/alenex/2010/alx10_005_birnm.pdf

See also: 11th Workshop on Algorithm Engineering and Experiments (ALENEX 2010)

Balanced Search Trees Simplified

Siddhartha Sen (Princeton University, US)

We present new balanced search trees that are simpler and more efficient in several ways, showing that the design space of this classical data structure is far from being exhausted.

We begin with the rank-balanced tree, a simple relaxation of AVL trees that takes $O(1)$ amortized time to rebalance after insertions and deletions.

Rank-balanced trees perform fewer rotations than red-black trees and achieve better height bounds. Using a new analysis that relies on an exponential potential function, we show that rebalancing modifies nodes exponentially infrequently in their heights.

Building on these techniques, we show that balanced search trees remain efficient even if deletion is done without any rebalancing. These results justify the practice of many B-tree-based database systems. In the case of binary trees, the underlying data structure must be modified to obtain such a result, leading to the relaxed AVL (ravl) tree. Ravl trees have height logarithmic in the number of insertions, and rebalancing modifies nodes exponentially infrequently in their heights. However, this seems to require $\Omega(\log\log n)$ bits of balance information at each of the n nodes.

Both rank-balanced trees and ravl trees show good promise in practice.

Keywords: Balanced search tree, red-black tree, data structure, exponential potential function

Joint work of: Haeupler, Bernhard; Sen, Siddhartha; Tarjan, Robert E.

Data Structures for Range-Closest-Pair Queries

Michiel Smid (Carleton University - Ottawa, CA)

Given a set S of n points in the plane, we want to construct a data structure that supports queries of the following type: Given a query range Q, report the closest pair in the set $Q \cap S$. We consider the cases when Q is a half-plane or an axes-parallel rectangle.

The basic approach is to consider the graph of the answers to all possible queries. If Q is a half-plane, a vertical strip, or a quadrant, this graph is sparse. Using this property, we can map the problem to that of finding the shortest edge in the graph that is completely inside Q.

If Q is a half-plane, this results in a data structure of size $O(n \log n)$ having query time $O(n^{1/2+\epsilon})$.

If Q is an axes-parallel rectangle, by combining the results for strips and quadrants with range-tree-like data structures, we obtain query time $O(\log^2 n)$ using space $O(n \log^5 n)$.

In order to efficiently build these data structures, we show that the semi-separated pair decomposition can be used to compute a supergraph of the graph of all possible answers; this supergraph has size $O(n \log n)$ and can be constructed in $O(n \log n)$ time.

Keywords: Range searching, closest-pair, spanners, semi-separated pair decomposition

Joint work of: Smid, Michiel; Gupta, Prosenjit; Janardan, Ravi; Kumar, Yokesh; Abam, Mohammad; Carmi, Paz; Farshi, Mohammad

Rank-Pairing Heaps

Robert E. Tarjan (Princeton University, US)

We describe a heap (priority queue) data structure with amortized performance matching that of Fibonacci heaps ($O(1)$ time for find-min, insert, meld, and decrease-key; and $O(\log n)$ time for delete) and simplicity rivaling that of pairing heaps. The structure uses integer node ranks to guarantee efficiency. A decrease-key operation takes $O(1)$ restructuring time worst-case and $O(1)$ rank changes amortized.

Keywords: Heap, priority queue, data structure, amortized efficiency

Joint work of: Haeupler, Bernhard; Sen, Siddhartha; Tarjan, Robert E.

External memory data structures with o(1)-I/O updates

Ke Yi (The Hong Kong Univ. of Science & Technology, CN)

I will survey the known results, both upper and lower bounds, on external memory data structures that supports updates in o(1) I/Os (for sufficiently large B) while still being able to answer queries quickly. Then some interesting open problems will be discussed.