BRICS Research Activities
Algorithms

Gerth Stølting Brodal

BRICS
Basic Research
in Computer Science

Outline

- The Algorithms Group
- ALCOM FT
- Upcoming Algorithm Events
- Algorithm Expertise within BRICS
- CCI Europe
- Dynamic Convex Hull
- External Memory Algorithms
- Maps On Us

The Algorithms Group


Sven Skyum
Algorithms, Complexity Theory


Erik Meineche Schmidt
Algorithms, Complexity Theory


Ivan Bjerre Damgaard
Cryptology


Peter Bro Miltersen
Complexity Theory, Data Structures


Gudmund Skovbjerg Frandsen
Algebraic Algorithms, Dynamic Algorithms


Christian Nørgaard Storm Pedersen
Bioinformatics, String Algorithms


Gerth Stølting Brodal
Data Structures, External Memory


Rolf Fagerberg
Data structures, External Memory


Mary Cryan
Learning of Distributions


Anders Yeo
Graph Theory, Graph Algorithms


Peter Høyer
Quantum Computations

PhD students


Jakob Pagter
Time-Space Trade-Offs


Riko Jacob
Optimization, Computational Geometry


Rasmus Pagh
Data Structures, Hashing


Alex Rune Berg
Graph Theory

Jesper Makholm Nielsen
Complexity Theory

Bjarke Skjernaa
Algorithms

ALCOM-FT

BRICS
Basic Research
in Computer Science

Algorithms and Complexity – Future Technologies

The ALCOM-FT project is a joint effort between ten of the leading groups in algorithms research in Europe. The aim of the project is to discover new algorithmic concepts, identify key algorithmic problems in important applications, and contribute to the accelerated transfer of advanced algorithmic techniques into commercial systems.

The project takes place from June 2000 to June 2003. It is supported by the European Commission under the Information Society Technologies programme of the Fifth Framework, as project number IST-1999-14186.

- ALCOM-FT is a continuation of ALCOM, ALCOM-II, ALCOM-IT
- BRICS is the coordinator of ALCOM-FT

BRICS
Basic Research
in Computer Science

ALCOM-FT Sites

BRICS	Erik Meineche Schmidt
Barcelona	Josep Díaz
Cologne	Michael Jünger
INRIA Rocquencourt	Philippe Flajolet
Max-Planck-Institut für Informatik	Kurt Mehlhorn
Paderborn	Burkhard Monien Friedhelm Meyer auf der Heide
Patras	Paul Spirakis
Rome "La Sapienza"	Giorgio Ausiello
Utrecht	Jan van Leeuwen
Warwick	Mike Paterson

Upcoming Algorithm Events

August 28–31, 2001

ESA 2001 – 9th Annual European Symposium on Algorithms
WAE 2001 – 5th Workshop on Algorithm Engineering

Summer 2002

Summer school on "External Memory Algorithms"

Ongoing

Alcom seminar

Algorithm Expertise within BRICS

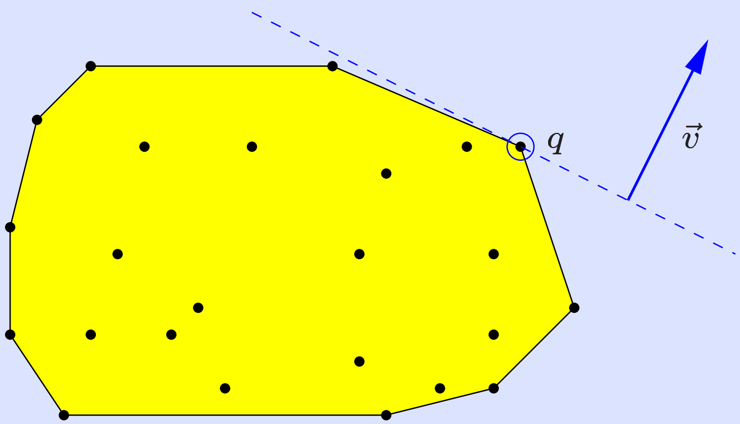
- Algorithms in general
- Data structures
- Dynamic algorithms
- External memory algorithms
- Algorithm engineering / experimental algorithmics

CCI Europe

- Domain specific project
- "Automatic layout of
  JyllandsPosten's JobSection"
- Problem ≈ 2D bin packing
- 2–approximation ?
- Enumeration + heuristics
- BRICS people
  - Kristian Høgsberg
  - Riko Jacob
  - Anders Yeo
  - Gerth Stølting Brodal
  - Erik Meineche Schmidt

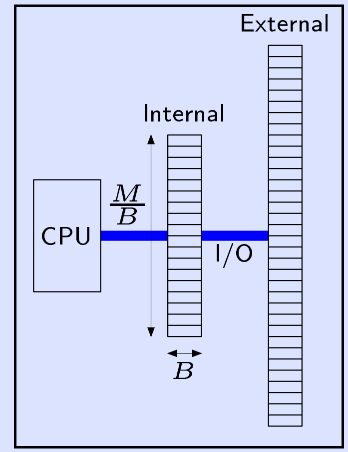
Dynamic Convex Hull

Insert(p), Delete(p)	Insert/delete a point p
Query($\vec{v}$)	Find extreme point on CH in direction $\vec{v}$



	Overmars van Leeuwen 1981	Chan 1999	Brodal Jacob 2000
Updates	$O(\log^2 n)$	$O(\log^{1+\varepsilon} n)$	$O(\log n \cdot \log\log n)$
Queries	$O(\log n)$	$O(\log n)$	$O(\log n)$

External Memory Algorithms



I/O model [Aggarwal and Vitter]

M = Internal memory size

N = Problem size

B = Block size

Complexity = # block I/Os to solve a problem

Examples

$\text{Scan}(N) = O(\frac{N}{B})$

$\text{Sort}(N) = O(\frac{N}{B} \cdot \log_{M/B} \frac{N}{M})$

Minimum Spanning Tree (MST)

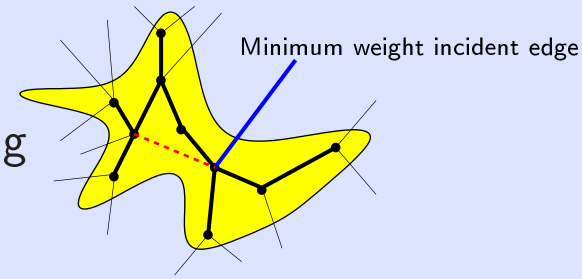
Compute the MST of a weighted graph with V vertices and E edges

Internal Chazelle 1999	External Arge, Brodal, Toma 2000
$O(E \cdot \alpha(E, V))$	$O(\text{Sort}(E) \cdot \log\log \frac{V \cdot B}{E})$

Minimum Spanning Tree

Prim's algorithm

Grow a single tree by iteratively including a minimum weight incident edge

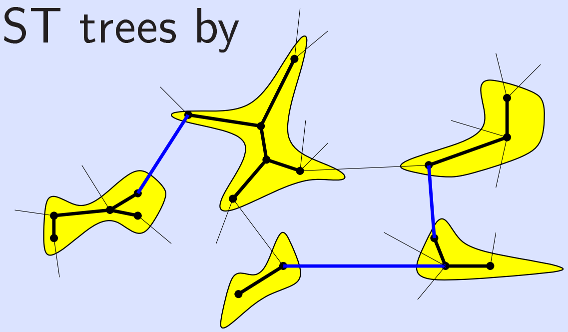


Priority queue on incident edges

- Internal $O(E \cdot \log E)$
- External $O(\underbrace{V}_{\text{buttleneck if } V \cdot B = \Omega(E)} + \text{Sort}(E))$

Kruskal's algorithm

In $O(\log V)$ phases grow independent MST trees by picking minimum weight incident edges

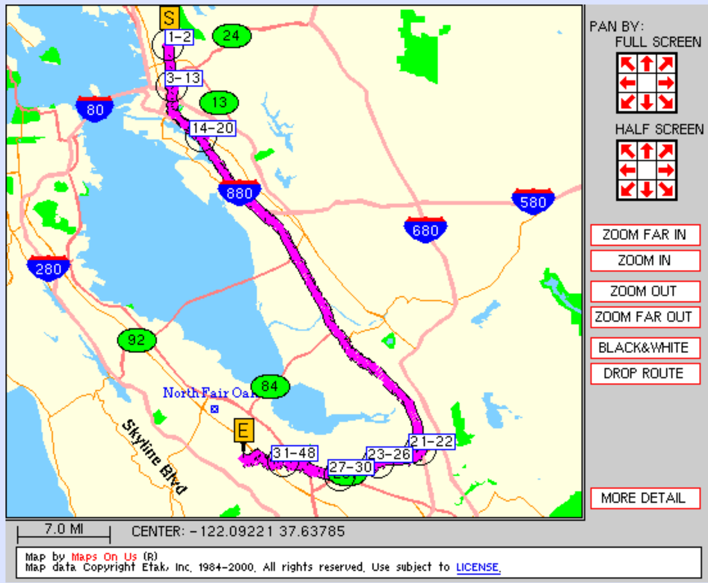


- Internal $O(E \cdot \log V)$
- External $O(\text{Sort}(E) \cdot \log V)$

Using "superphases" $V \rightarrow \frac{V}{k}$ requires $O(\text{Sort}(E) \cdot \log \log k)$ I/Os

Let $k = \frac{V \cdot B}{E}$ and switch to Prim implies the external result $\square$

BRICS
Basic Research
in Computer Science



"However, because of the size of the routing data, we have to use heuristics when planning routes (i.e., we find "close to optimum" routes rather than optimum routes). As a result, sometimes a Favor Highways route will be slightly faster than the Fastest route. This is particularly true for routes longer than about 100 miles. ...
Our routing will continually improve as the quality of our data improves and as we invent better routing algorithms."