BRICS Research Activities
Algorithms

Gerth Stølting Brodal

Outline of Talk

- The Algorithms Group
- Courses
- Algorithm Events
- ALCOM FT
- Expertise within BRICS

Examples

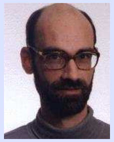
- Algorithms for Evolutionary Trees
- Cache Oblivious Algorithms

The Algorithms Group


Erik Meineche Schmidt
Algorithms, Complexity Theory


Sven Skyum
Algorithms, Complexity Theory


Peter Bro Miltersen
Complexity Theory, Data Structures

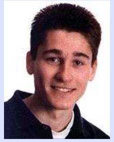

Gudmund Skovbjerg Frandsen
Algebraic Algorithms, Dynamic Algorithms


Christian Nørgaard Storm Pedersen
Bioinformatics, String Algorithms


Gerth Stølting Brodal
Data Structures, External Memory


Rolf Fagerberg
Data structures, External Memory

PhD students


Alex Rune Berg
Graph Theory


Jesper Makholm Byskov
Algorithms for NP-hard problems


Bolette Ammitzbøll Madsen
Algorithms for NP-hard problems


Bjarke Skjernaa
Algorithms for NP-hard problems

Kristoffer Arnsfelt Hansen
Complexity

Courses

- dProg (Introduction to Programming) *Frandsen*
- dADS (Algorithms and Data Structures) *Brodal, Schmidt*
- dSøgOpt (Searching and Optimization) *Skyum, Miltersen*
- Computer Architecture and Operating Systems *Pedersen*
- Computational Geometry *Skyum*
- Complexity *Miltersen*
- Randomized, Parallel and Dynamic Algorithms *Frandsen*
- Algorithms (ph.d. course) *Brodal, Fagerberg*
- External Memory Algorithms and Data Structures —
- Algorithms for Web Indexing and Searching —
- Algorithms in Bioinformatics *Pedersen*
- Genome Analysis —

Algorithm Events

Upcoming

18th IEEE Conference on Computational Complexity
(June 2003, organizing chair Peter Bro Miltersen)

Ongoing

Alcom seminar

Recent

EEF Summer School on Massive Data Sets (June 2002)

ALGO 2001 (August 2001)

- ESA 2001 - 9th Annual European Symposium on Algorithms
- WAE 2001 - 5th Workshop on Algorithm Engineering
- WABI 2001 - 1st Workshop on Algorithms in BioInformatics

Algorithms and Complexity – Future Technologies

The ALCOM-FT project is a joint effort between eleven 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.

- ALCOM-FT (continuation of ALCOM, ALCOM-II, ALCOM-IT)
- BRICS is the coordinator of ALCOM-FT
  (Erik Meineche Schmidt, Rolf Fagerberg)
- 336 technical reports since July 2000

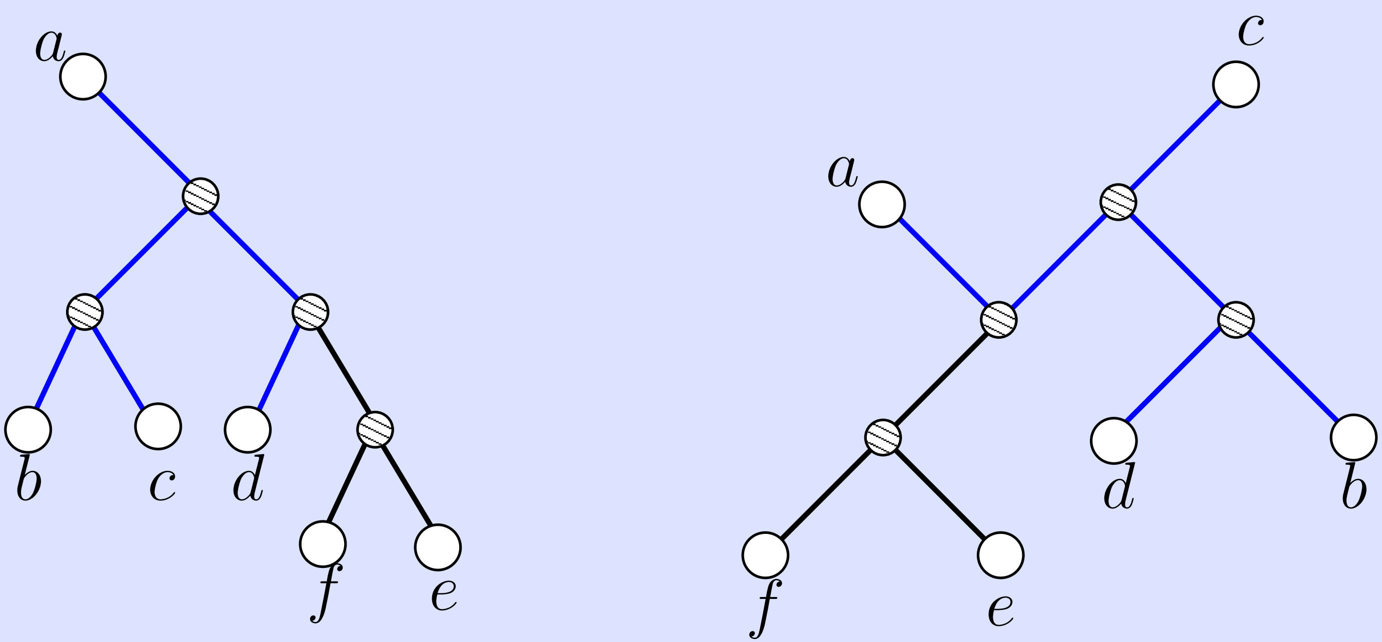
BRICS	Erik Meineche Schmidt
Barcelona	Josep Díaz
Cologne	Michael Jünger
INRIA Rocquencourt	Philippe Flajolet
Max-Planck-Institut für Informatik	Kurt Mehlhorn
Paderborn	Burkhard Monien
	Friedhelm Meyer auf der Heide
Patras	Paul Spirakis
Rome “La Sapienza”	Giorgio Ausiello
Utrecht	Jan van Leeuwen
Warwick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

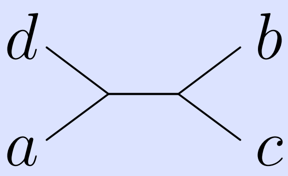
Algorithm Expertise within BRICS

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

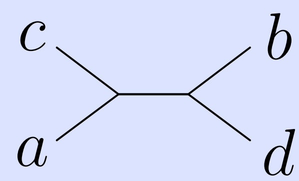
Examples

Algorithms for Evolutionary Trees



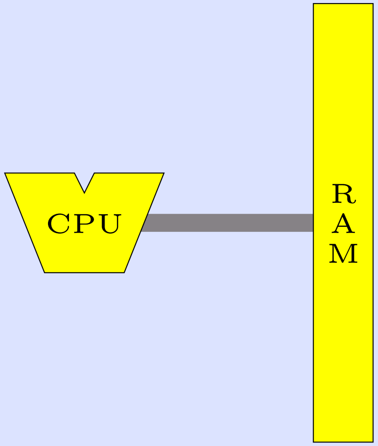


induced quartets



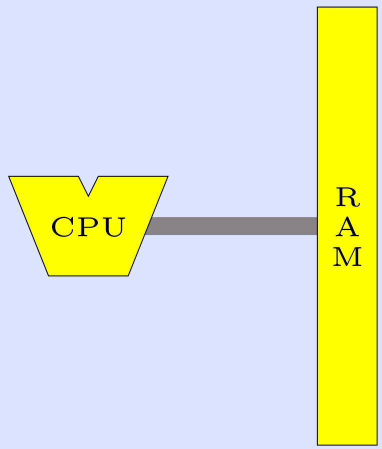
- Reconstruct evolutionary trees, e.g. from quartets
- Compare evolutionary trees, e.g. number of common quartets

The Classic RAM Model



Add:	$O(1)$
Mult:	$O(1)$
Mem access:	$O(1)$

The Classic RAM Model

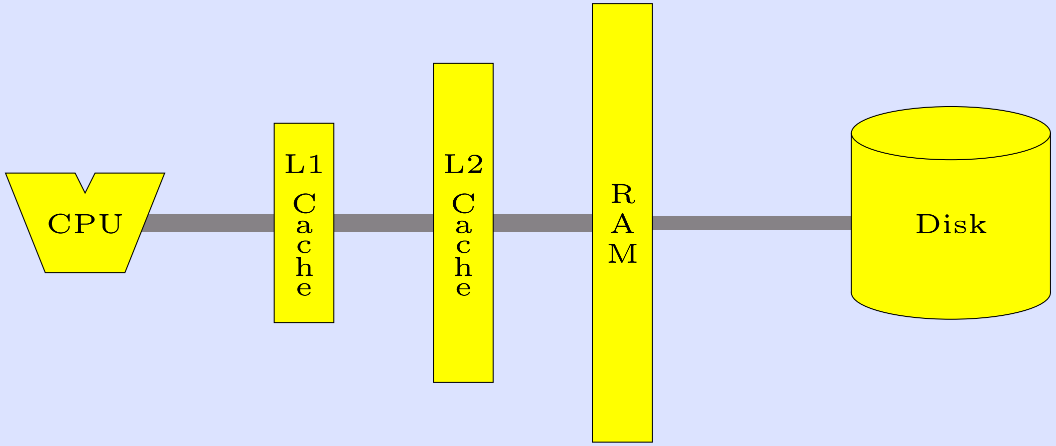


Add: $O(1)$

Mult: $O(1)$

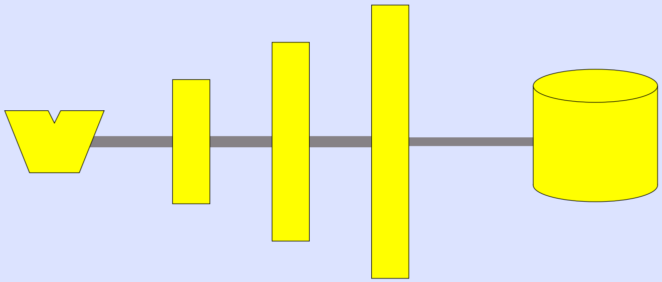
~~Mem access: $O(1)$~~

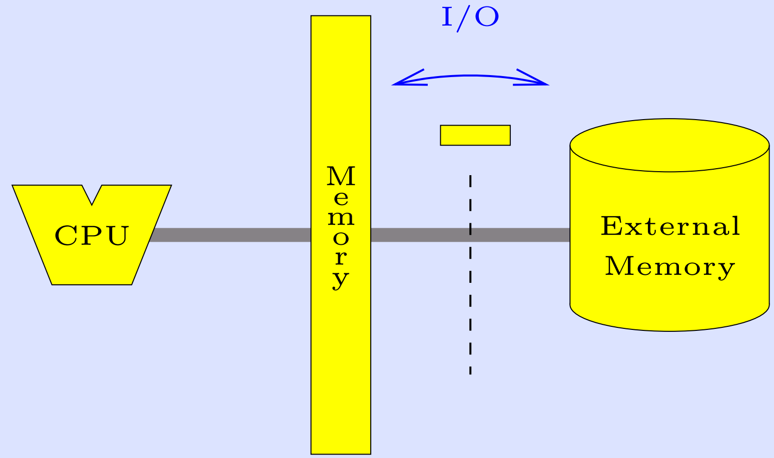
Real life



Bottleneck: transfer between two highest memory levels in use

The I/O Model





N = problem size

M = memory size

B = I/O block size

Aggarwal and Vitter 1988

- One I/O moves B consecutive records from/to disk
- Cost: number of I/Os
- E.g., sorting requires $O\left(\frac{N}{B}\log_{M/B}\frac{N}{B}\right)$ I/Os.

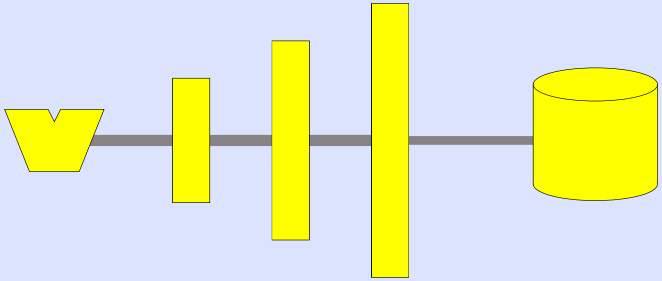
Cache-Oblivious Model

Frigo, Leiserson, Prokop, Ramachandran, FOCS'99

- Program in the RAM model
- Analyze in the I/O model (for arbitrary B and M).

Advantages

- Optimal on arbitrary level $\Rightarrow$ optimal on all levels.
- B and M not hard-wired into algorithm.



Cache-Oblivious Results

- Scanning $\Rightarrow$ stack, queue, median finding,. . . .
- Sorting, matrix multiplication, FFT

  Frigo, Leiserson, Prokop, Ramachandran, FOCS'99

- Cache oblivious search trees

  Prokop 99
  Bender, Demaine, Farach-Colton, FOCS'00
  Rahman, Cole, Raman, WAE'01
  Bender, Duan, Iacono, Wu and Brodal, Fagerberg, Jacob, SODA'02

- Priority queue and graph algorithms

  Arge, Bender, Demaine, Holland-Minkley, Munro, STOC'02
  Brodal, Fagerberg, ISAAC'02

- Computational geometry

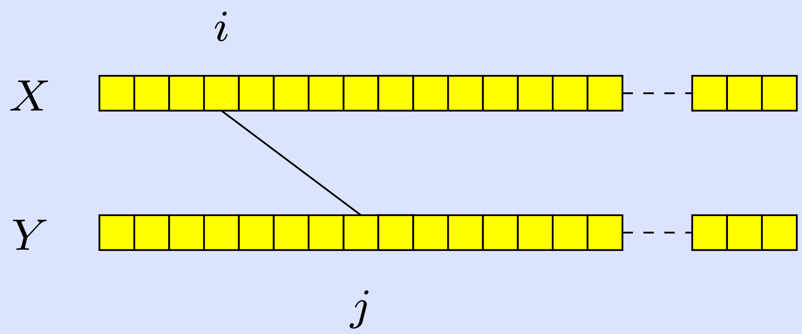
  Bender, Cole, Raman, ICALP'02
  Brodal, Fagerberg, ICALP'02

- Scanning dynamic sets

  Bender, Cole, Demaine, Farach-Colton, ESA'02

Double for-loop

X, Y arrays of length n:



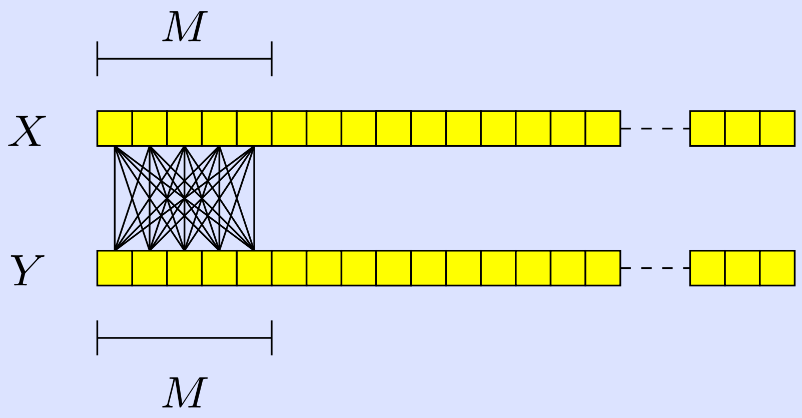

```
for(i=0; i<n; i++)
   for(j=0; j<n; j++)
      f(X[i], Y[j])
```

I/O complexity:

$$n \times \frac{n}{B} = \frac{n^2}{B}$$

Double for-loop

More I/O-efficient version:



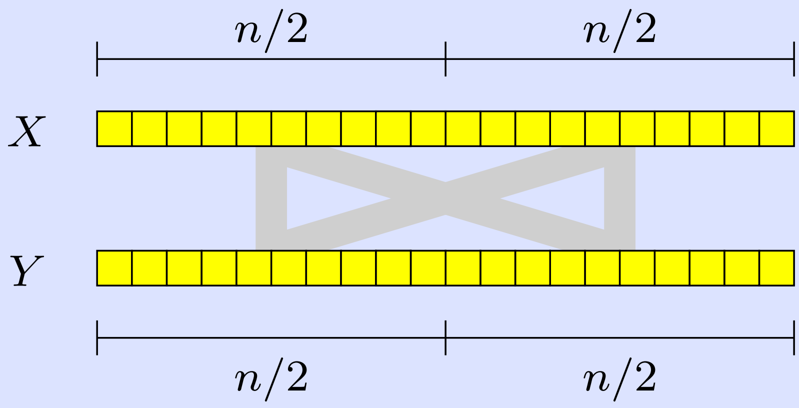
I/O complexity:

$$\frac{n}{M} \times \frac{n}{M} \times \frac{M}{B} = \frac{n^2}{MB}$$

```
for(i=0; i<n; i=i+M)
   for(j=0; j<n; j++)
      for(k=i; k<i+M; k++)
         f(X[k], Y[j])
```

Double for-loop

Cache-oblivious version:



+ recursion

I/O complexity:

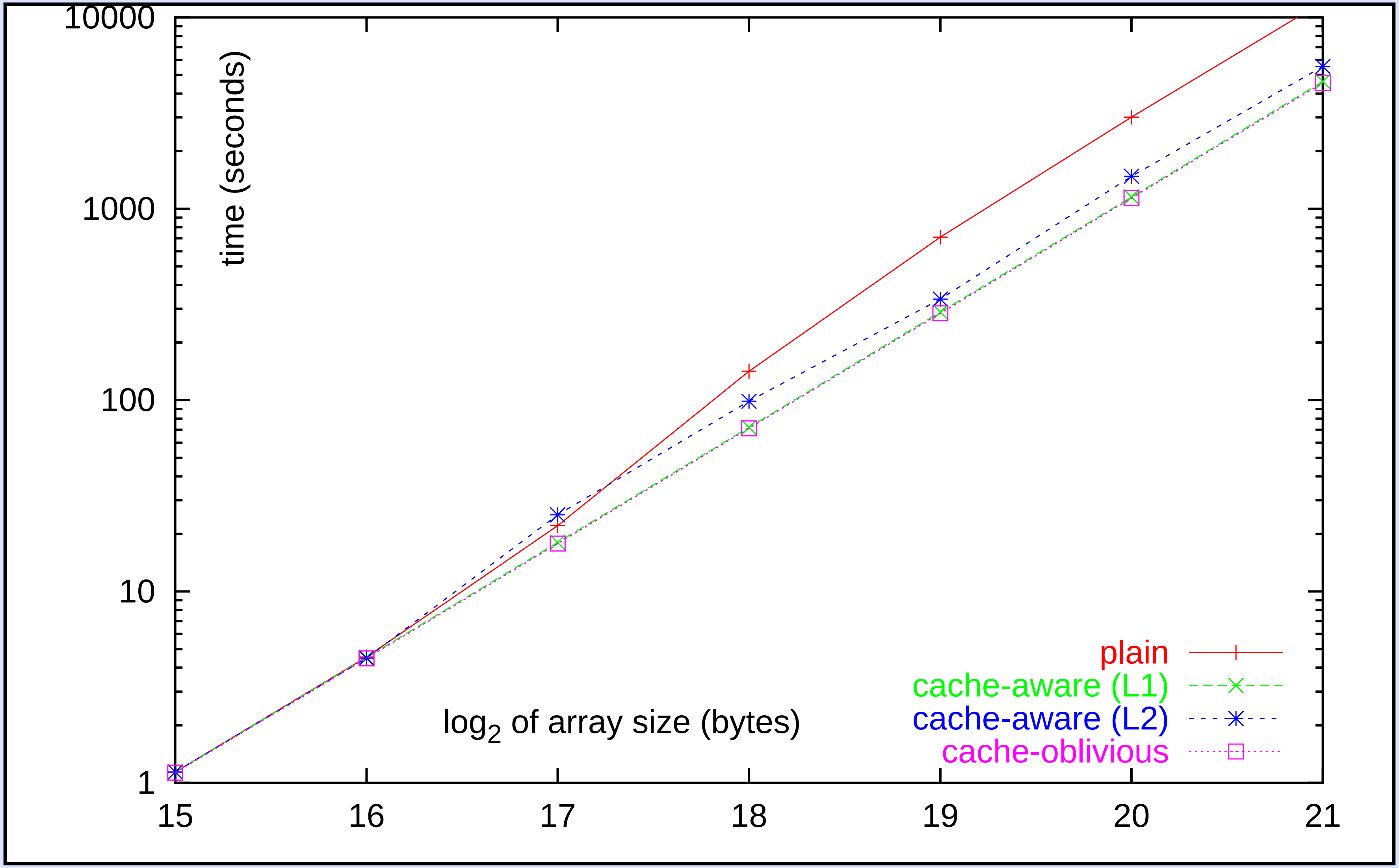
Again $\dfrac{n^2}{MB}$

Double for-loop

Cache-oblivious version

```
doubleLoop(i, j, length)
   if (length == 1)
      f(X[i], Y[j])
   else
      doubleLoop(i, j, length/2)
      doubleLoop(i, j+length/2, length/2)
      doubleLoop(i+length/2, j, length/2)
      doubleLoop(i+length/2, j+length/2, length/2)
```

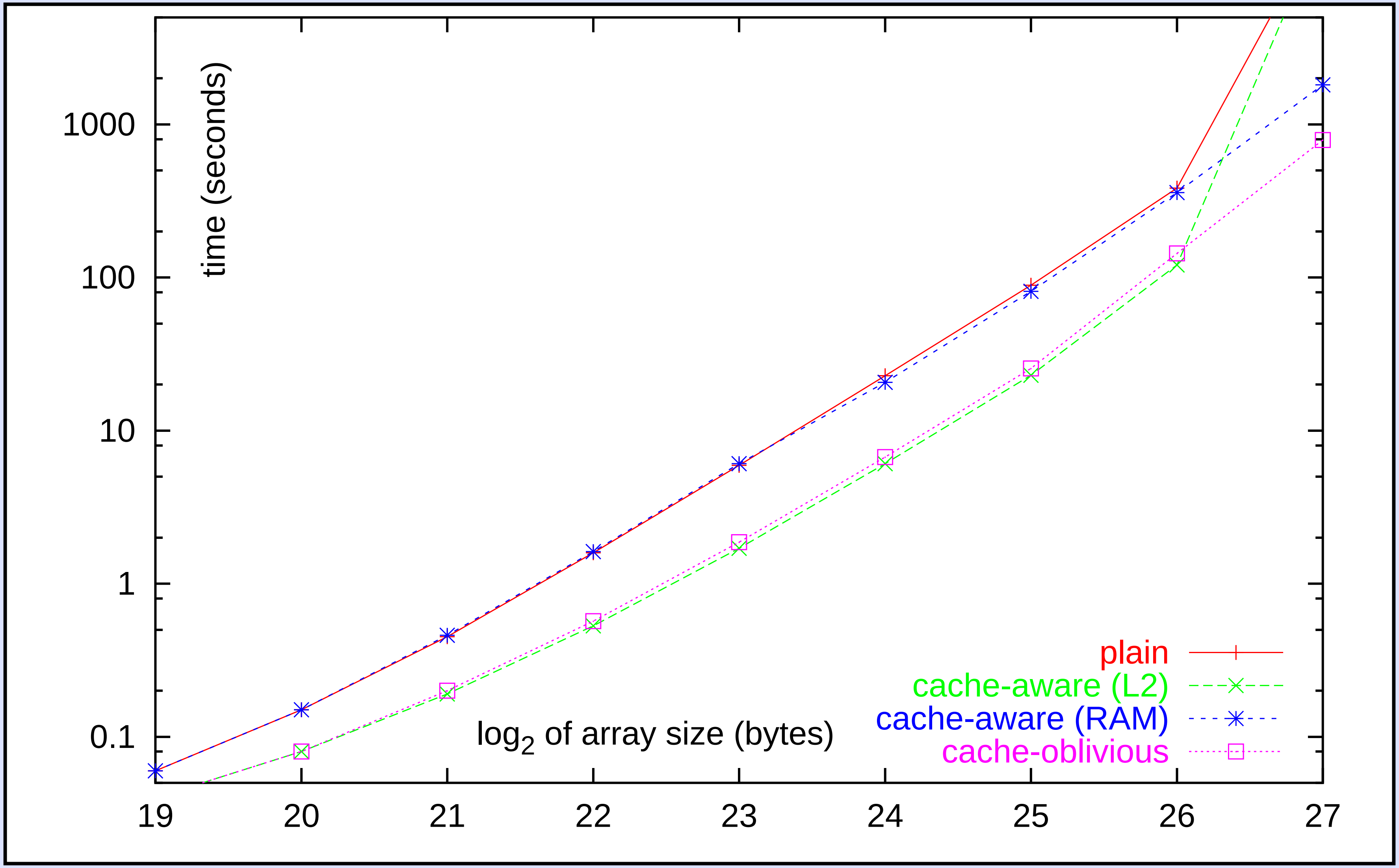
Double for-loop – Experiments



Sizes within RAM (element size 4 bytes)

366 MHz Pentium II, 128 MB RAM, 256 KB Cache, gcc -O3, Linux

Double for-loop – Experiments



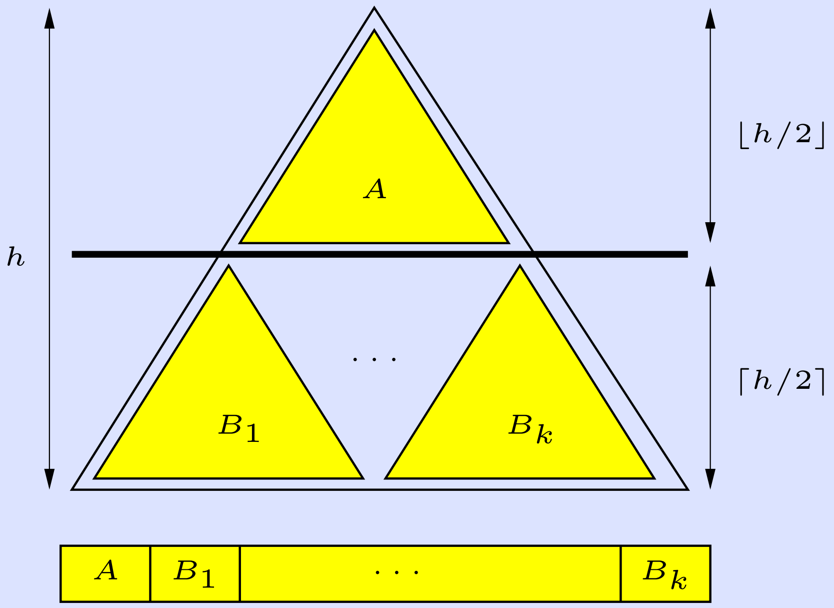
Sizes exceeding RAM (element size 1 KB)

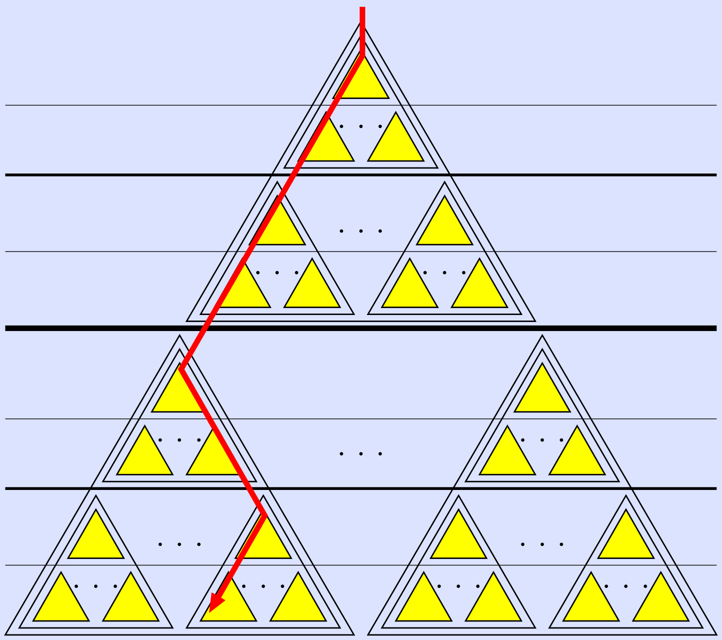
366 MHz Pentium II, 128 MB RAM, 256 KB Cache, gcc -O3, Linux

Cache-Oblivious Search Trees

Recursive memory layout (van Emde Boas layout)

Prokop 1999





Binary tree

Searches use $O(\log_B N)$ I/Os

Bender, Demaine, Farach-Colton, FOCS'00

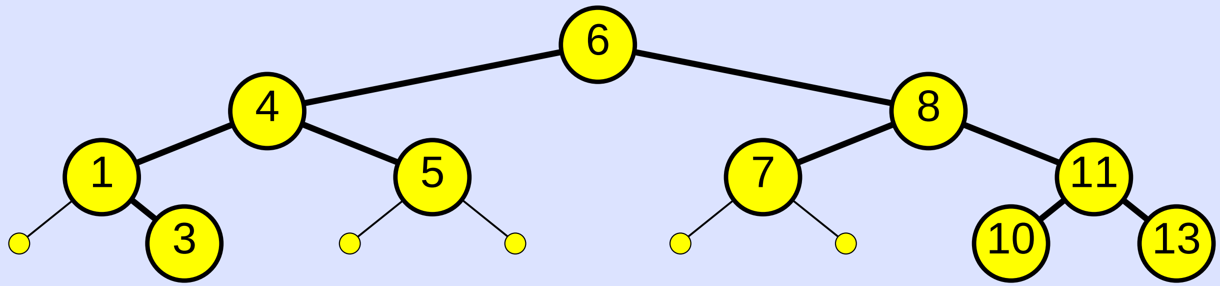
Rahman, Cole, Raman, WAE'01

Bender, Duan, Iacono, Wu, SODA 02

Brodal, Fagerberg, Jacob, SODA'02

Dynamization

Cache-Oblivious Search Trees



$$\Downarrow$$

6	4	8	1	–	3	5	–	–	7	–	–	11	10	13

Search	$O(\log_B N)$
Range Reporting	$O\left(\log_B N + \frac{k}{B}\right)$
Updates	$O\left(\log_B N + \frac{\log^2 N}{B}\right)$

Current Work... $M \geq B^2$?