

Algorithms and Data Structures  
for Faulty Memory

# Computer Science Day 2008

Gerth Stølting Brodal



# madALGO

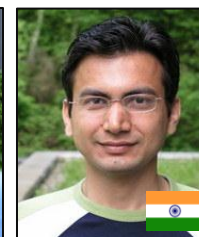
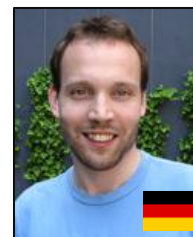
CENTER FOR MASSIVE DATA ALGORITHMICS



## Faculty



## Post Docs



## Administration



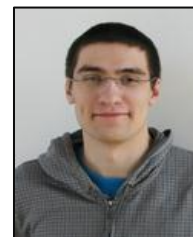
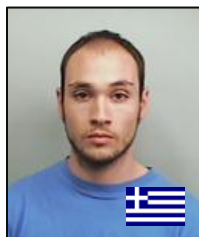
## PhD students



## Master Students



## Programmers

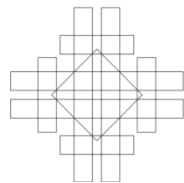




August 18-21, 2008

## MADALGO Summer School on Cache-Oblivious Algorithms

June 8-10, 2009



## 25th Annual ACM Symposium on Computational Geometry



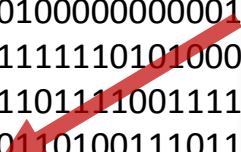
*"You have to provide reliability on a software level. If you're running 10,000 machines, something is going to die every day."*

—  fellow Jeff Dean

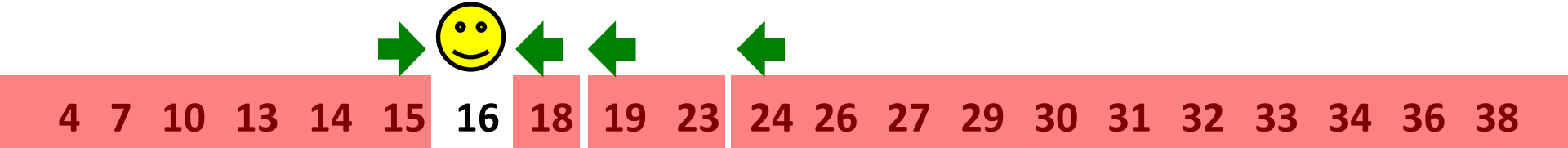
101100011011100111011001010100101010100111110010000100011110010111100001000111001111010001010001110  
100001001110000100001111011100110111101010110011101011110100111000010000111110110100110001000011011  
011100101011111111110000100101011000001110110110101110011001001100001011011111011011011011010100000  
000000010001100001000011101001110010100100100000001010000000100010010001101010011100100110001000010  
00101111111010010011001100001001100110100101000000010011010011101001001101100011100010011101001111  
11011010011110100001101010010011100110111110000101001010100010001001011011101111110000110100100001  
00001011000101110101001111111001100111111010010100010001001100010100100110000001000001100011101011  
110000110010111000100111101000001101011110010001101001000101000111101001000011010001011000110101011  
110010101100010010010110101010100001010100000110110000000101011001011101110001001011111101010110111  
110000100001010011010101111000111110011010100111011101001011011000100011011011111010001000100110110  
010011110111010110011011111110100000001111010000101110010011100011010100001100100110001111101100010  
110000011101100110100111100010001110001001111010001000110101010001010110110101000011001001000110101  
1111111110110110110011110010101000001100011101010111001010101100101001011100101001011011011010110000101011  
100011001101001100010100001000000000011110010011010100100111111010010000011100111001110010001000000  
101101101111101000010101111111101010000101101000010110011100111010011001000100111111011101010110111  
010101010100010011001111111011110011111110011101011110100001100100001010010010010101011110011100010  
010111101101000001001001101101001110111011101011000010000111011000011101000011011010110110110110010  
11010110101001101110000001010010111101001000101010010101111101010111111110010001101100110001000110  
00111010110001111000101010110110111111110111011000111101000110100000000001111110011101110000001110  
100001011011100110100010010111110000010100000011000000000010110001010010110011101001001101101110110  
10111100111001010000111100101000100001101010110010100000011011110011111000110101100000010111000001  
111000011010011011111010010101011110110010100111001010110111101000001000011010110101100110001011010  
110101001101000111010101000110100100001001110000011010111101011100010111101000011011000010101000110  
011110101101001111011000010110001010011011000101101100100110010100010101101011101100011011101010111  
1111100110101100000000000000001000001000001100110001101101100100011001101001011010110011000000000101  
010100111011100100101011110001011001010100010011100001000000111001100000110100110001000000001100111  
001010000010111001001100111000110011100011011000011010001111011100101011101001101100001010011010110  
110011101011010111110100111010010011111110111001000100100010101011110111011101001100010000010110110  
100000110011011000011110110000010110101111110100011100000110001111111000001011010100100001011000010  
000010100001000100000001011011110010101011101000010100010001100000101010101010100110010111011101011  
110001011000001100111010011011011111010111101011011000100001110110011101000010100111110010010101101  
01101001111100000010101110001011011000111111001110100010001100001100011000100101010101001

101100011011100111011001010100101010100111110010000100011110010111100001000111001111010001010001110  
100001001110000100001111011100110111101010110011101011110100111000010000111110110100110001000011011  
01110010101111111111000010010101100000111011011010111001100100110000101101111101101101101101010000  
000000010001100001000011101001110010100100100000001010000000100010010001101010011100100110001000010  
001011111111010010011001100001001100110100101000000010011010011101001001101100011100010011101001111  
11011010011110100001101010010011100110111110000101001010100010001001011011101111110000110100100001  
00001011000101110101001111111001100111111010010100010001001100010100100110000001000001100011101011  
110000110010111000100111101000001101011110010001101001000101000111101001000011010001011000110101011  
110010101100010010010110101010100001010100000110110000000101011001011101110001001011111101010110111  
110000100001010011010101111000111110011010100111011101001011011000100011011011111010001000100110110  
0100111101110101100110111111101000000011101001001101011010110101101011010110101111101100010  
1100000111011001101001111000100011100010011111010001000110101010001010110110101000011001001000110101  
1111111101101101100111100101010000011000111111010110000100011010110101101011010110000101011  
100011001101001100010100001000000000011100100110101001001111110100100000111001110011001000100000  
1011011011111010000101011111111010100001011100001100111001111001101000100111111011101010110111  
01010101010001001100111111101111110011111110011101011000011001000011001000010100101110011100010  
010111101101000001001001101101001110111011101011000010000111011000011101000011011010110110110010  
110101101010011011100010101001011101001000101011110100100010101111011001110100011011001000110  
001110101100011110001010101101101111111110111011000111101000110100000000001111110011101110000001110  
100001011011100110100010010111110000010100000011000000000010110001010010110011101001001101101110110  
101111001110010100001111001010001000011010110010100000011011110011111000110101100000010111000001  
111000011010011011111010010101011110110010100111001010110111101000001000011010110101100110001011010  
110101001101000111010101000110100100001001110000011010111101011100010111101000011011000010101000110  
011110101101001111011000010110001010011011000101101100100110010100010101101011101100011011101010111  
1111100110101100000000000000001000001000001100110001101101100100011001101001011010110011000000000101  
010100111011100100101011110001011001010100010011100001000000111001100000110100110001000000001100111  
001010000010111001001100111000110011100011011000011010001111011100101011101001101100001010011010110  
110011101011010111110100111010010011111110111001000100100010101011110111011101001100010000010110110  
100000110011011000011110110000010110101111110100011100000110001111111000001011010100100001011000010  
000010100001000100000001011011110010101011101000010100010001100000101010101010100110010111011101011  
11000101100000110011101001101101111101011101011011000100001110110011101000010100111110010010101101  
0110100111110000001010111000101101100011111100011100110100010001100001100001100010101010101001

A bit in memory changed  
value because of e.g.  
background radiation,  
system heating, ...

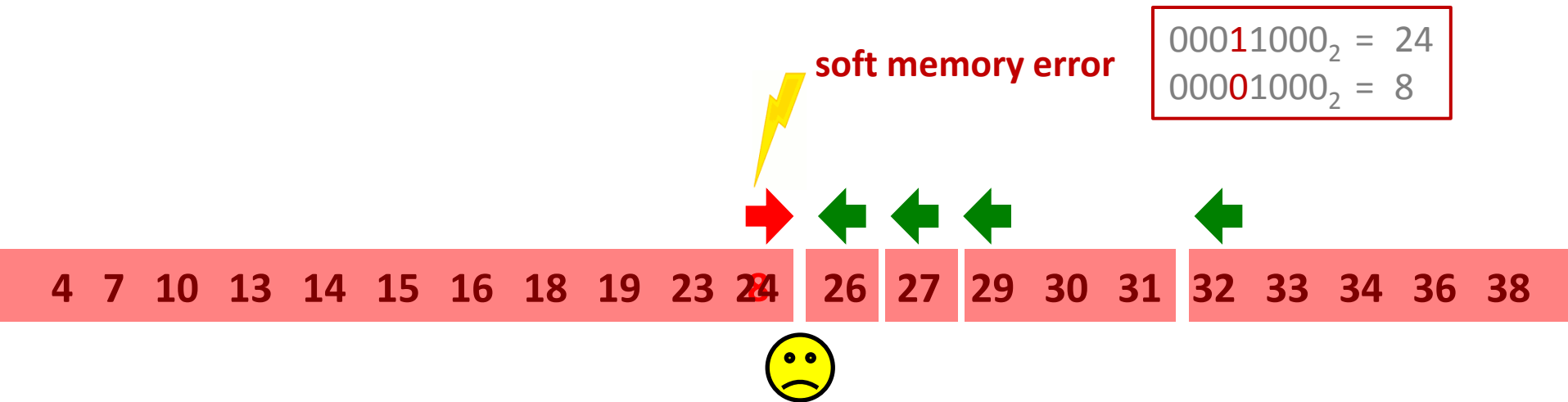


# Binary Search for 16



$O(\log N)$  comparisons

# Binary Search for 16



**Requirement:** If the search key occurs in the array as an uncorrupted value, then we should report a match !



# Where is Kurt ?



# Where is Kurt ?



# Where is Kurt ?

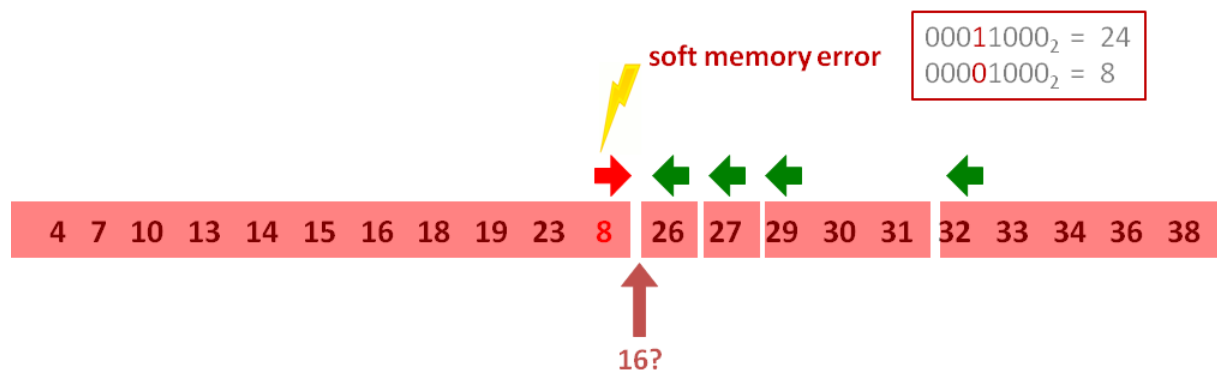


If at most 4 faulty answers then Kurt is somewhere here

# Faulty-Memory RAM Model

Finocchi and Italiano, STOC'04

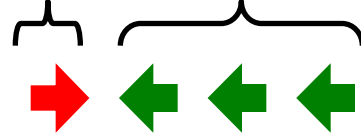
- **Content** of memory cells can get **corrupted**
- Corrupted and uncorrupted content **cannot be distinguished**
- $O(1)$  **safe** registers
- **Assumption:** At most  $\delta$  corruptions



# Faulty-Memory RAM: Searching

Problem?

Low confidence    High confidence



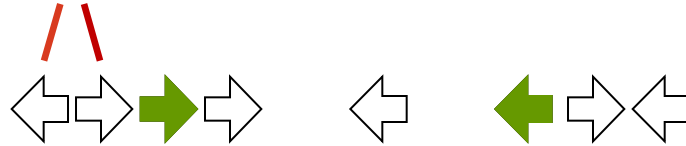
4 7 10 13 14 15 16 18 19 23 8 26 27 29 30 31 32 33 34 36 38

↑  
16?

# Faulty-Memory RAM: Searching

When are we  
done ( $\delta=3$ )?

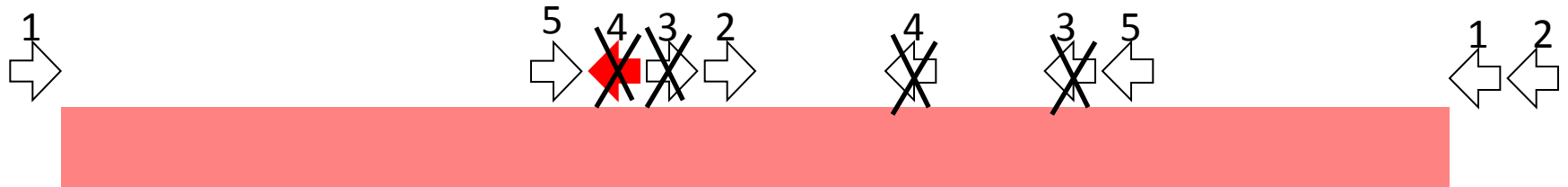
Contradiction, i.e. at  
least one fault



If range contains at least  $\delta+1 \rightarrow$  and  $\delta+1 \leftarrow$  then there is  
at least one uncorrupted  $\rightarrow$  and  $\leftarrow$ , i.e.  $x$  must be  
contained in the range

# Faulty-Memory RAM: $\Theta(\log N + \delta)$ Searching

Brodal, Fagerberg, Finocchi, Grandoni, Italiano, Jørgensen, Moruz, Mølhave, ESA'07

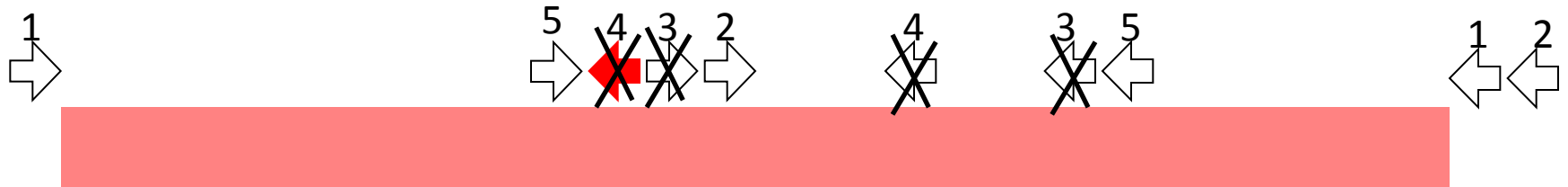


If verification fails

- contradiction, i.e.  $\geq 1$  memory-fault
- ignore 4 last comparisons
- **backtrack** one level of search

# Faulty-Memory RAM: $\Theta(\log N + \delta)$ Searching

Brodal, Fagerberg, Finocchi, Grandoni, Italiano, Jørgensen, Moruz, Mølhave, ESA'07



- Standard binary search + verification steps
- At most  $\delta$  verification steps can fail/backtrack
- **Detail:** Avoid repeated comparison with the same (wrong) element by grouping elements into blocks of size  $O(\delta)$



# Faulty-Memory RAM: Reliable Values

- Store  $2\delta+1$  copies of value  $x$  - at most  $\delta$  copies uncorrupted
- $x$  = majority
- Time  $O(\delta)$  using two safe registers (**candidate** and **count**)

Boyer and Moore '91

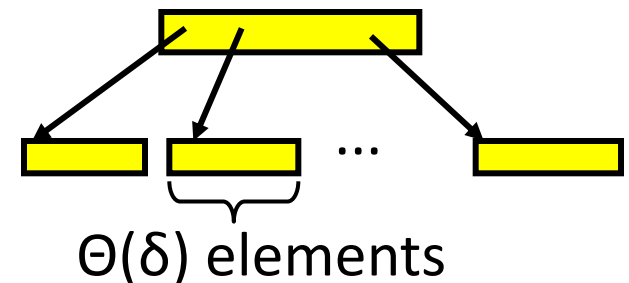
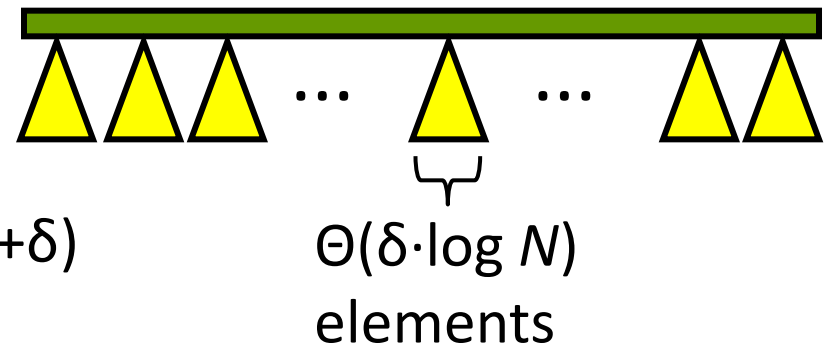
$\delta=5$	<table border="1"><tr><td><math>y</math></td><td><math>y</math></td><td><math>y</math></td><td><math>x</math></td><td><math>x</math></td><td><math>y</math></td><td><math>x</math></td><td><math>x</math></td><td><math>x</math></td><td><math>y</math></td><td><math>x</math></td></tr></table>	$y$	$y$	$y$	$x$	$x$	$y$	$x$	$x$	$x$	$y$	$x$
$y$	$y$	$y$	$x$	$x$	$y$	$x$	$x$	$x$	$y$	$x$		
<b>Candidate</b>	$y$ $y$ $y$ $y$ $y$ $y$ $y$ - $x$ - $x$											
<b>Count</b>	1 2 3 2 1 2 1 0 1 0 1											

# Faulty-Memory RAM: Dynamic Dictionaries

Brodal, Fagerberg,  
Fnocchi, Grandoni,  
Italiano, Jørgensen,  
Moruz, Mølhave,  
ESA'07

- Packed array
- Reliable pointers and keys
- Updates  $O(\delta \cdot \log^2 N)$
- Searches = fault tolerant  $O(\log N + \delta)$
- 2-level buckets of size  $O(\delta \cdot \log N)$
- Root: Reliable pointers and keys
- Bucket search/update amortized  $O(\log N + \delta)$
- Search and update amortized  $O(\log N + \delta)$

Itai, Konheim, Rodeh, 1981



# Fault-Tolerant Results

- Merging, time  $\Theta(N+\delta^2)$  [Finocchi, Grandoni, Italiano, ICALP'06](#)
- Priority queue, time  $\Theta(\log N+\delta)$  [Jørgensen, Moruz, Mølhave, WADS'07](#)
- Sorting, time  $\Theta(N \cdot \log N + \delta^2)$  [Finocchi, Grandoni, Italiano, ICALP'06](#)
- Static and dynamic dictionary, time  $\Theta(\log N + \delta)$   
[Brodal, Fagerberg, Finocchi, Grandoni, Italiano, Jørgensen, Moruz, Mølhave, ESA'07](#)  
[Finocchi, Grandoni, Italiano, ICALP'06](#)
- External-memory fault tolerant searching,  $\Theta\left(\frac{1}{\varepsilon} \log_B N + \frac{\delta}{B^{1-\varepsilon}}\right)$  I/Os  
[Brodal, Jørgensen, Mølhave, Submitted](#)



Allan G. Jørgensen and Gerth S. Brodal at MFCS'07