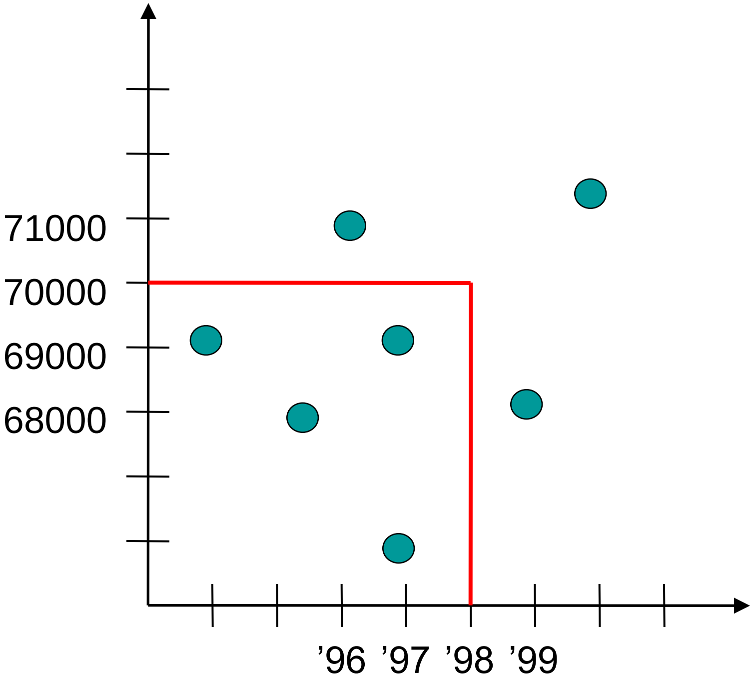
Lower Bounds for 2-Dimensional Range Counting

Mihai Pătraşcu

summer @ IBM

Range Counting

```
SELECT count(*)
FROM employees
WHERE salary <= 70000
  AND startdate <= 1998
```



Some Theory

- d dimensions
- range trees (roughly):
  space $S=n\lg^{d-1}n$, query $t=\lg^{d-1}n$
- space $S=n^{2d}$, query $t=O(1)$

- geometric extensions:
  range = disks, half-spaces, polygons…

PROBLEM: lower bounds

The Semigroup Industry

Let $(U,+)$ be a semigroup

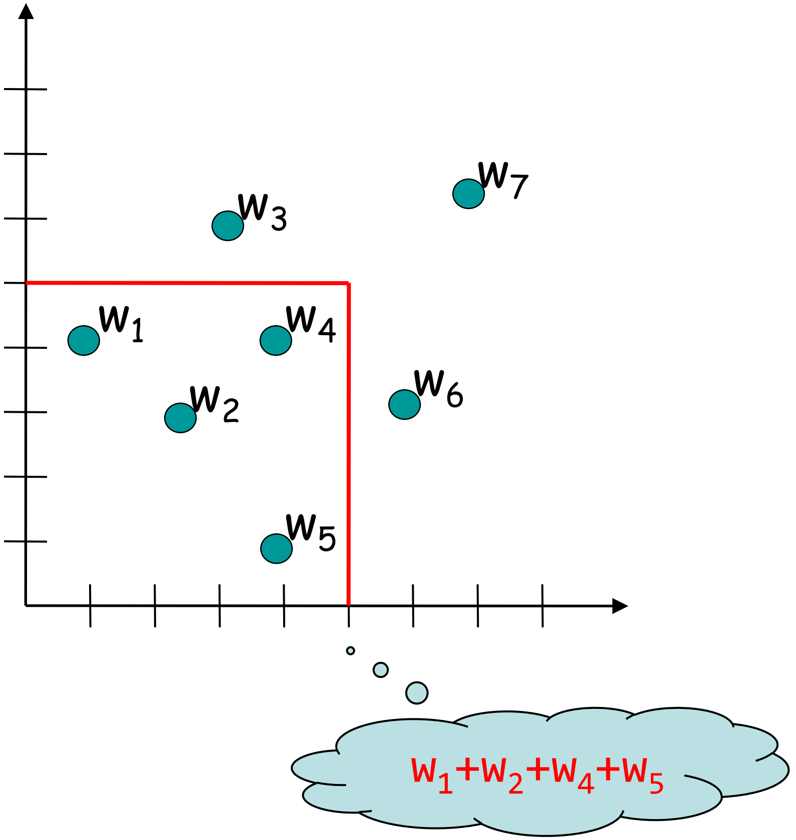
Points have weights in U

Given $w_1,\ldots,w_n$:

precompute S sums

query:

add t precomputed sums



Why “Industry”?

- tight semigroup lower bounds
  [Fredman JACM’81] [Chazelle FOCS’86]

- many, many excellent bounds for many range problems

[Caravaggio]

Semigroup = Low Dimension

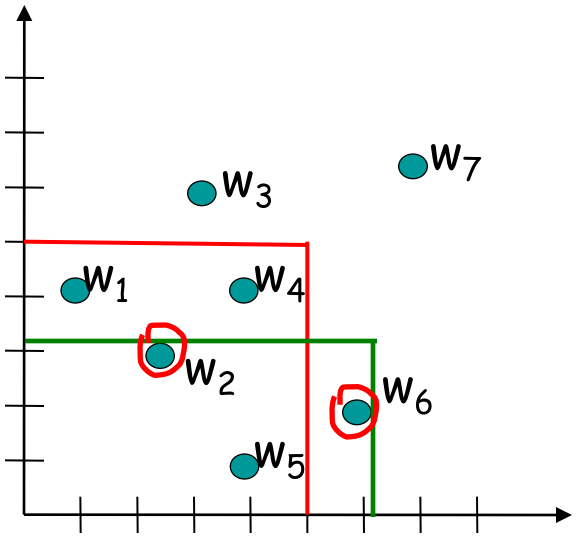
- say Mem[17]= w_2+w_6
- can Mem[17] help with this query?

  NO: cannot subtract w_6

- Mem[17] described by bounding box

  can only be used when

  query dominates bounding box

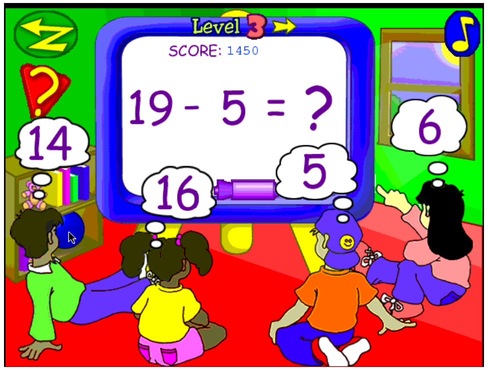


"How well do rectangles decompose into rectangles?"

X (disks, half-planes…) Y

All these objects are low-dimensional!

Computation = High Dimension



New concept…

weights come from a group (U,+,-)

- can cancel any weight => no bounding box
- relevant information about Mem[17]:
  O(1)-D rectangle → linear combination of n variables

- decomposability in O(1)-D → decomposability in n-D

- n-D means: geometry → information theory

The Ultimate Frontier

The cell-probe model:

- plain old counting (weights {0,1})
- each cell stores any function of inputs
- query probes t cells (adaptively), computes any function

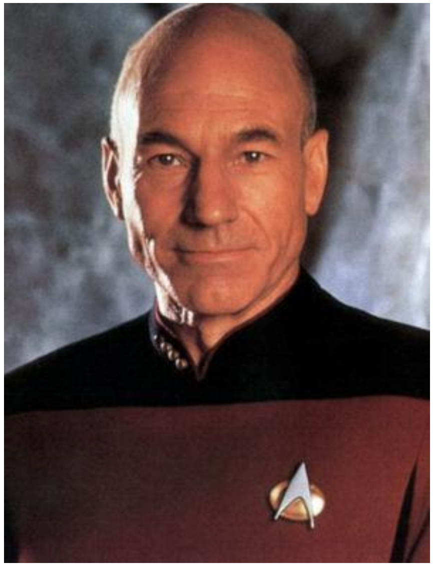
"Theory of the computer in your lap"

To boldly go where no one has gone before

General lack of group lower bounds

(never mind cell-probe!)

[Chazelle STOC'95] $\Omega(\lg\lg n)$



[Fredman JACM'82]
[Chazelle FOCS'86]
[Agarwal '9x, '0x]
etc etc

yeah, we have nice semigroup lower bounds
but prove something in the group model

Our Small Step

If $S=n\lg^{O(1)}n$, then $t=\Omega(\lg n/\lg\lg n)$

☺ group model!

☺ …and cell-probe model!

☺ tight in 2D

Maybe not so giant leap for mankind…

☹ does not grow with d

=> really only relevant for 2D

Basic Idea

Remember:
- space $S = n \lg^{d-1} n$, query $t = \lg^{d-1} n$
- space $S = n^{2d}$, query $t = O(1)$

Separa

[P,Thoru

[P,Thoru

[P,Thoru

ot tight)

NOW "trick" $\Omega(\lg n / \lg\lg n)$

(natural, fundamental problem; tight)

The trick:

"Consider $n/\lg n$ queries, see what happens"

This paper: what happens for range counting

Hard Instance

The *bit-reversal permutation* (see FFT, etc):

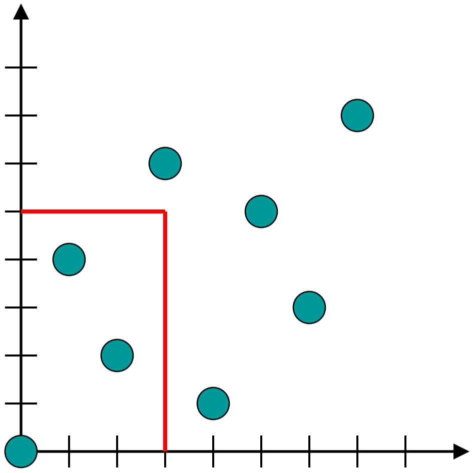
x	0	1	2	3	4	5	6	7
$\pi(x)$	0	4	2	6	1	5	3	7

e.g. $\pi(6) = \pi(\text{“110”}) = \text{“011”} = 3$

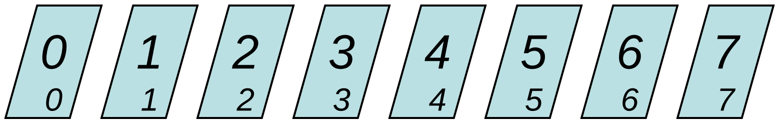
Well-known hard instance:

- points at $(x, \pi(x))$
- random query $[0,a] X [0,b]$

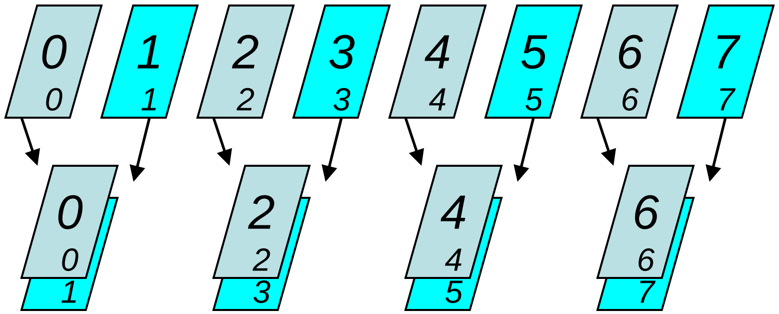
[in fact, many independent random queries]



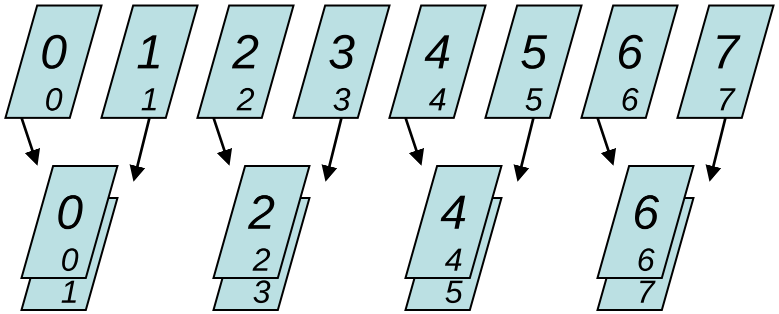
The Shuffle View



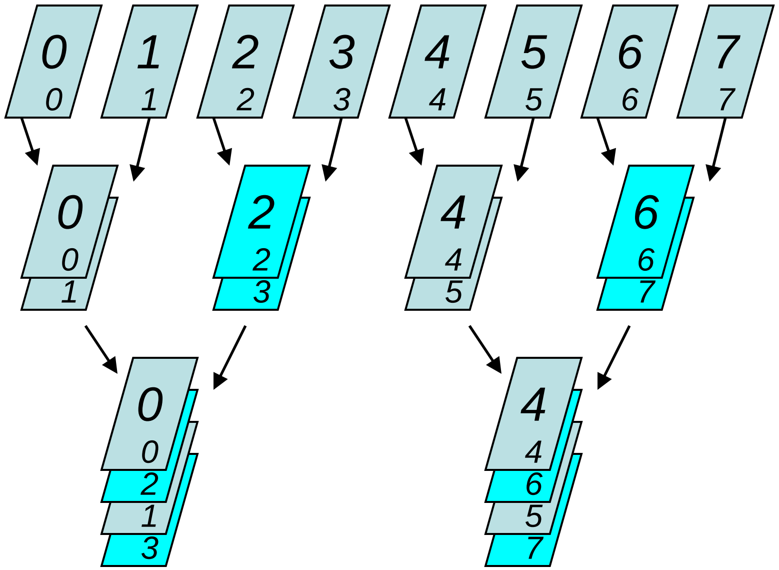
The Shuffle View



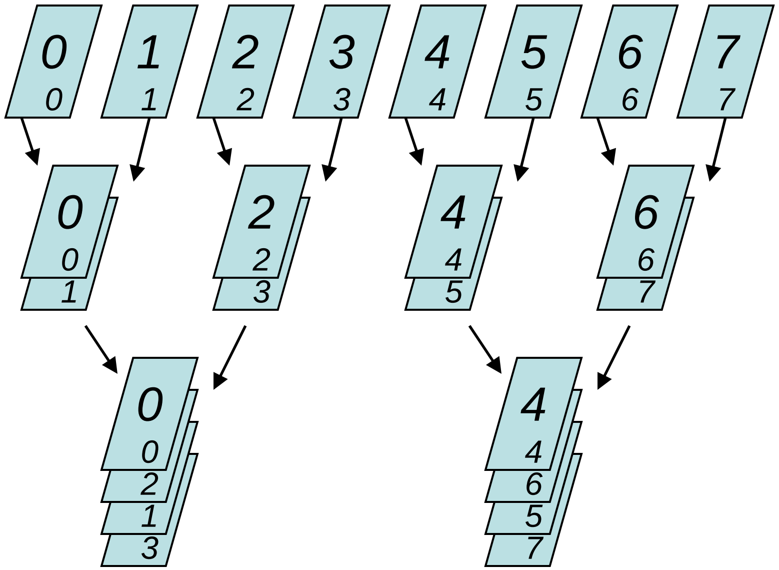
The Shuffle View



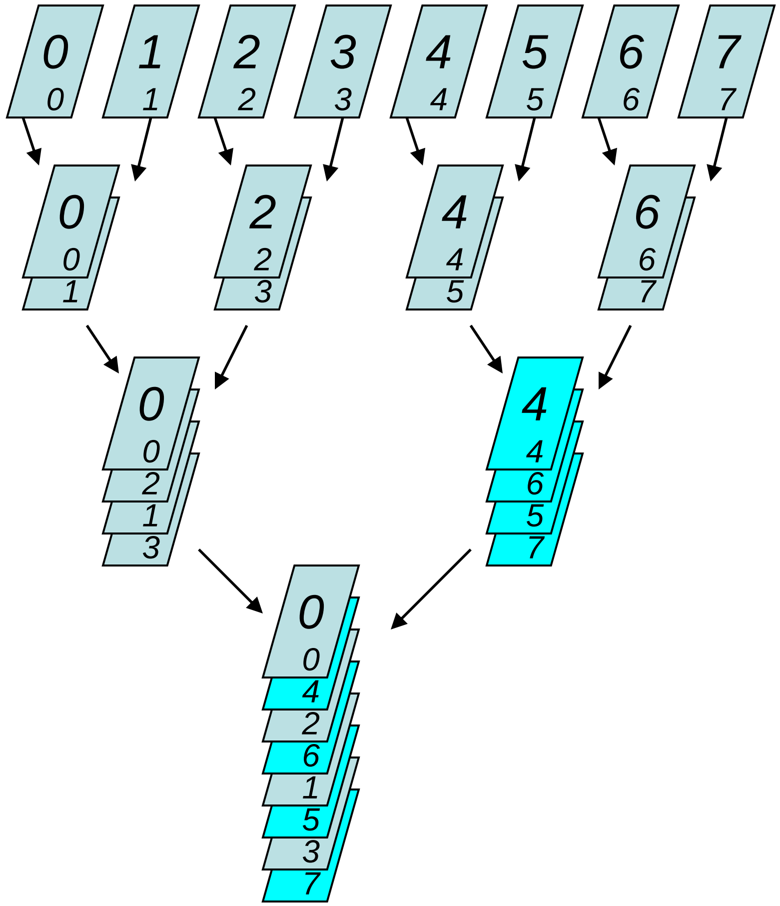
The Shuffle View



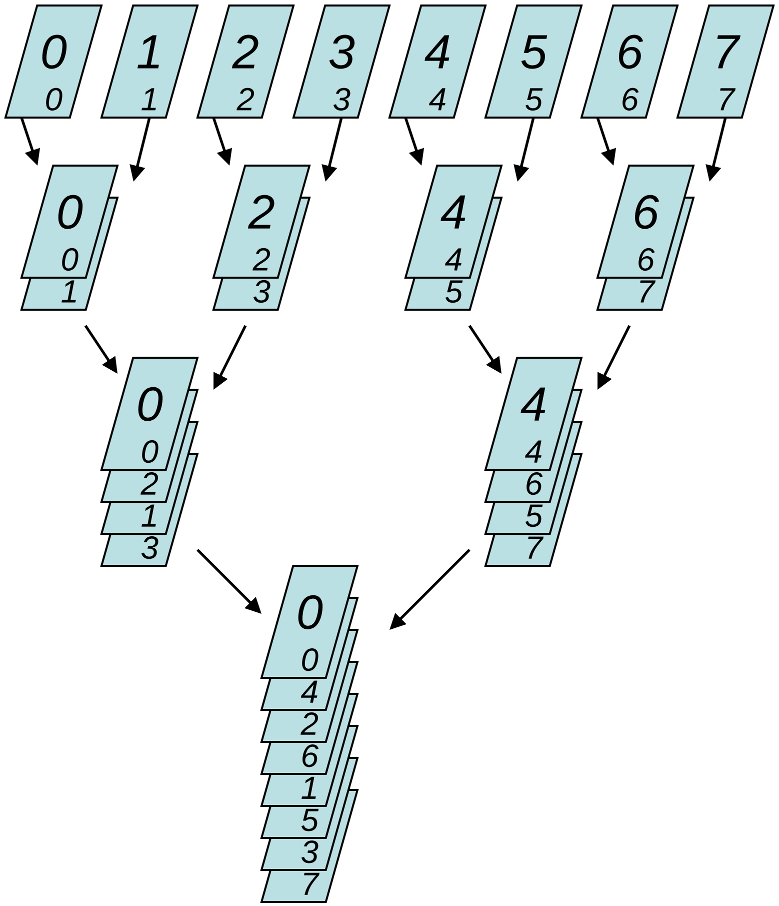
The Shuffle View

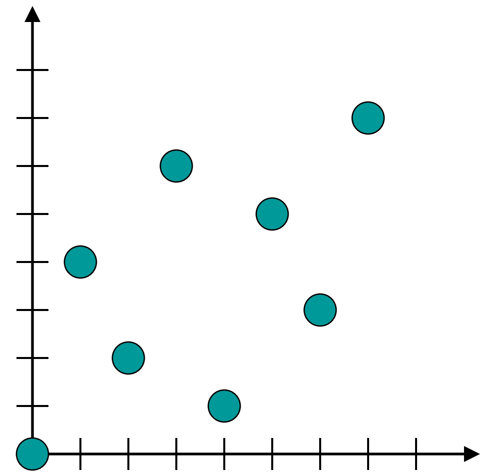


The Shuffle View



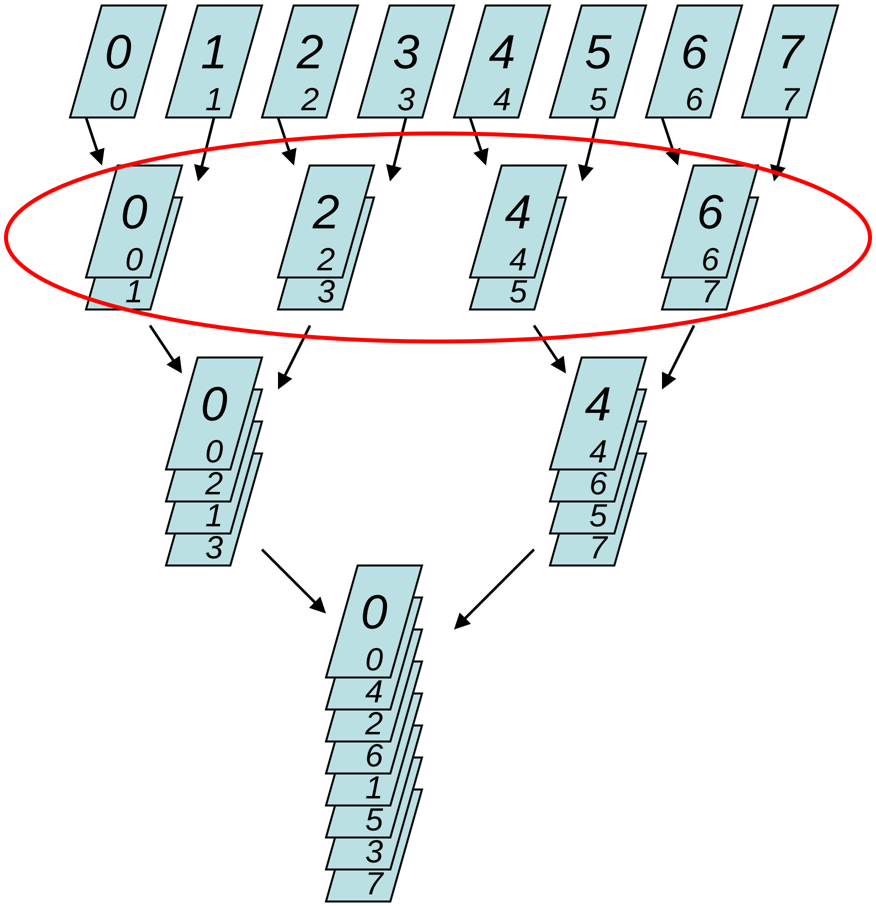
The Shuffle View

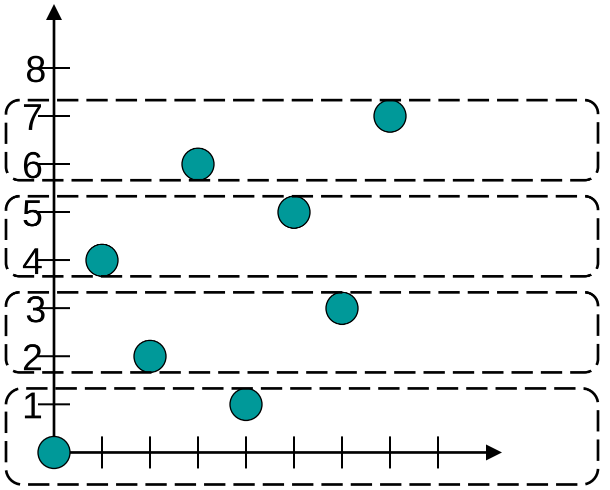




x	0	1	2	3	4	5	6	7
π(x)	0	4	2	6	1	5	3	7

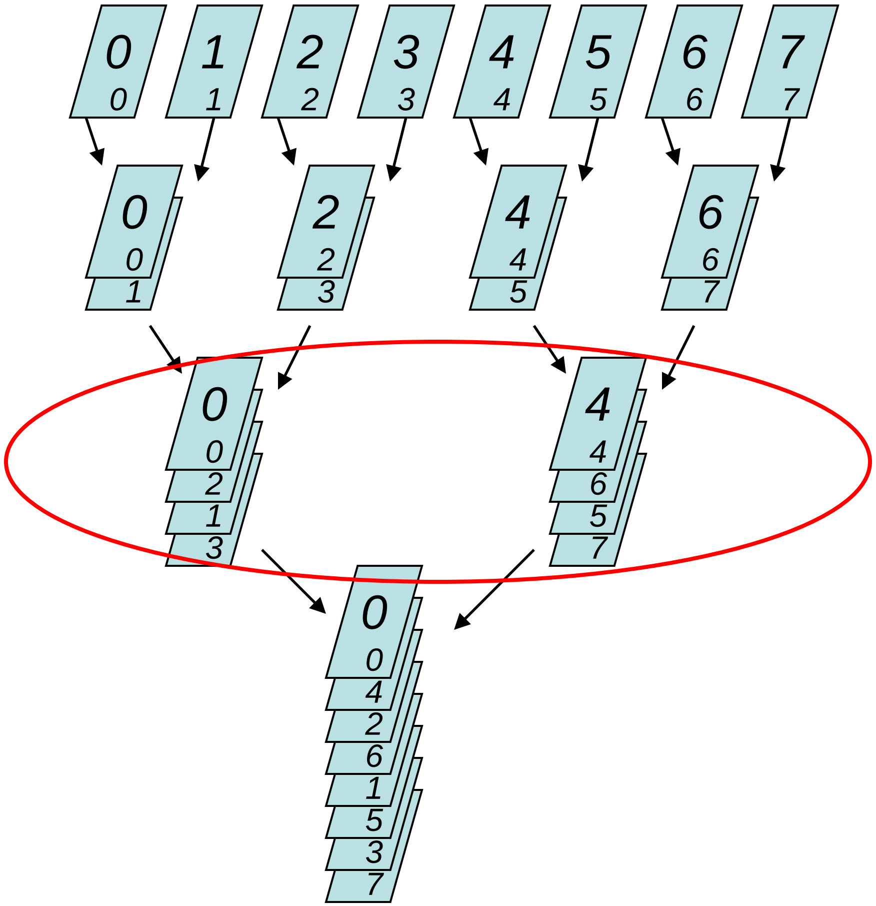
The Shuffle View

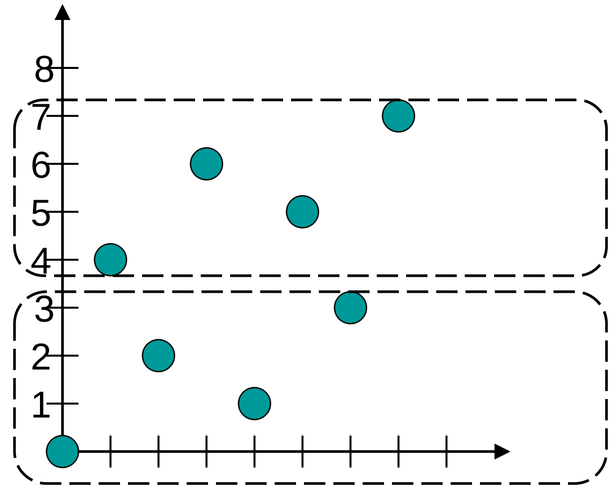




x	0	1	2	3	4	5	6	7
π(x)	0	4	2	6	1	5	3	7

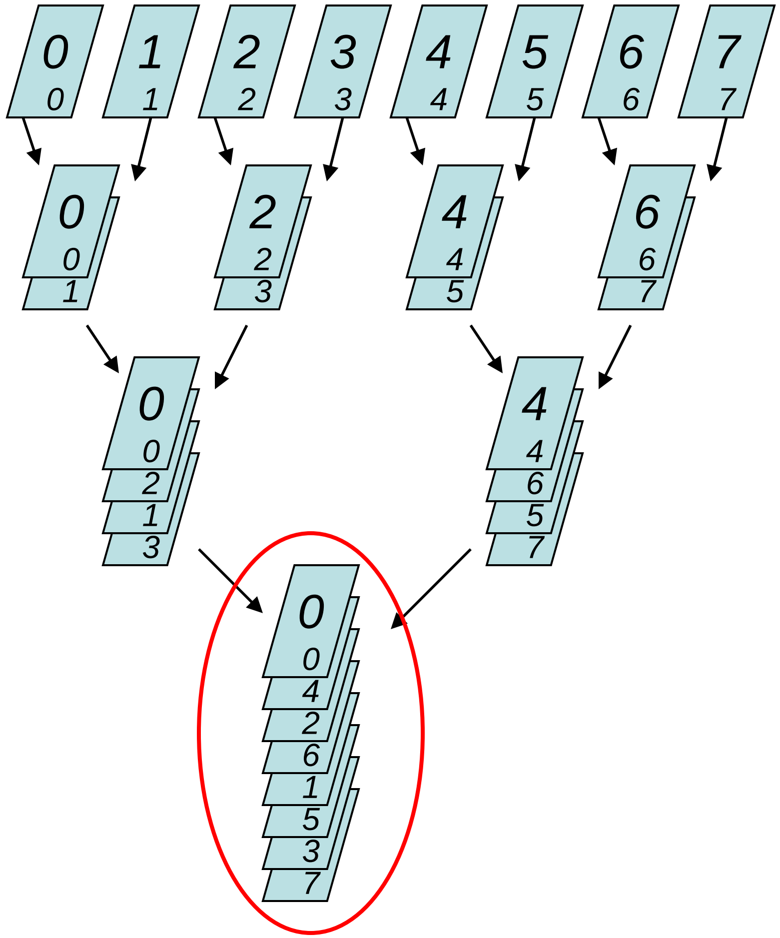
The Shuffle View

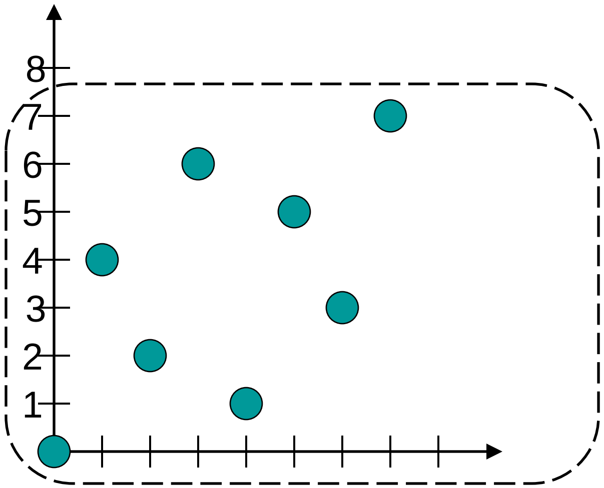




x	0	1	2	3	4	5	6	7
π(x)	0	4	2	6	1	5	3	7

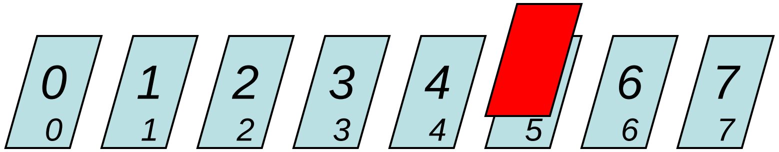
The Shuffle View



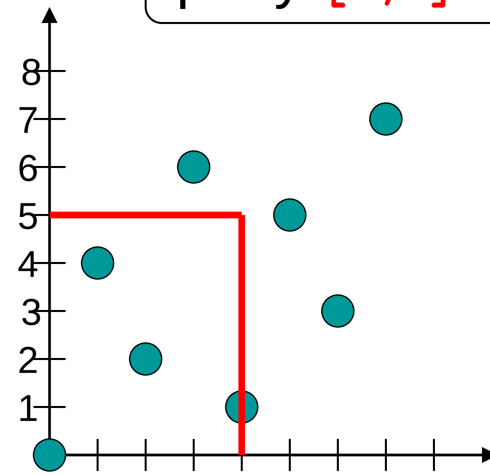


x	0	1	2	3	4	5	6	7
π(x)	0	4	2	6	1	5	3	7

Shuffling the Queries

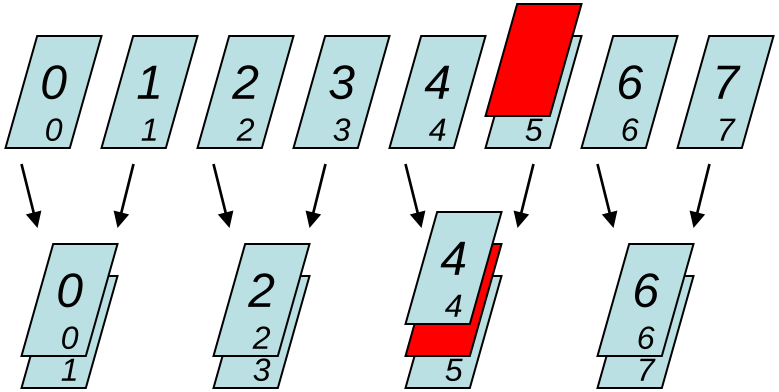


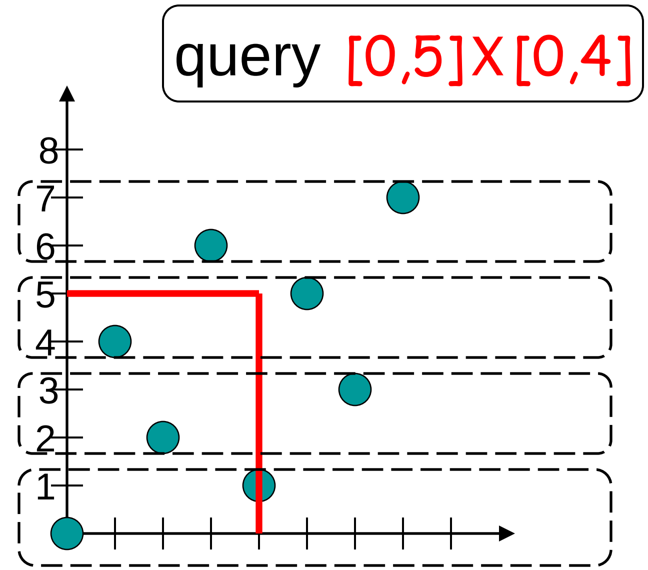
query [0,4]X[0,5]



x	0	1	2	3	4	5	6	7
π(x)	0	4	2	6	1	5	3	7

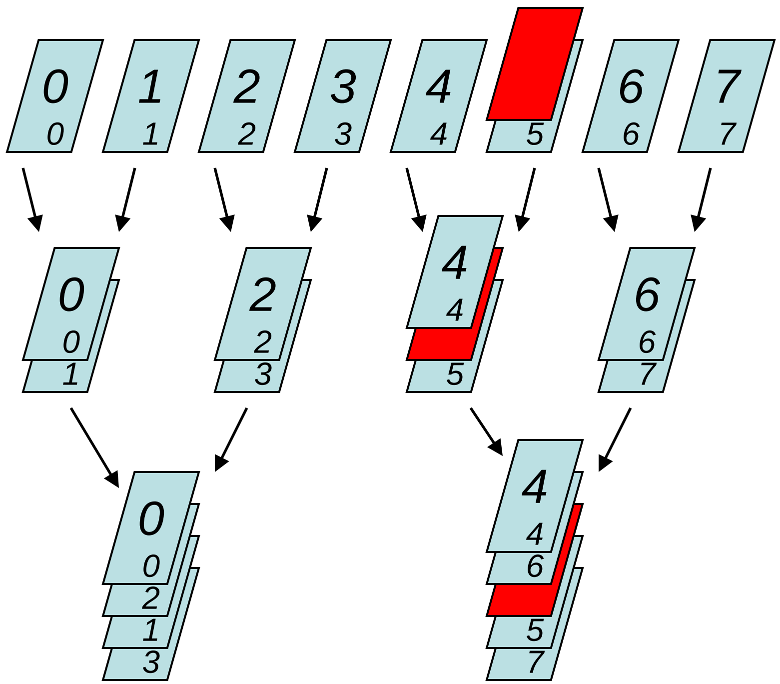
Shuffling the Queries

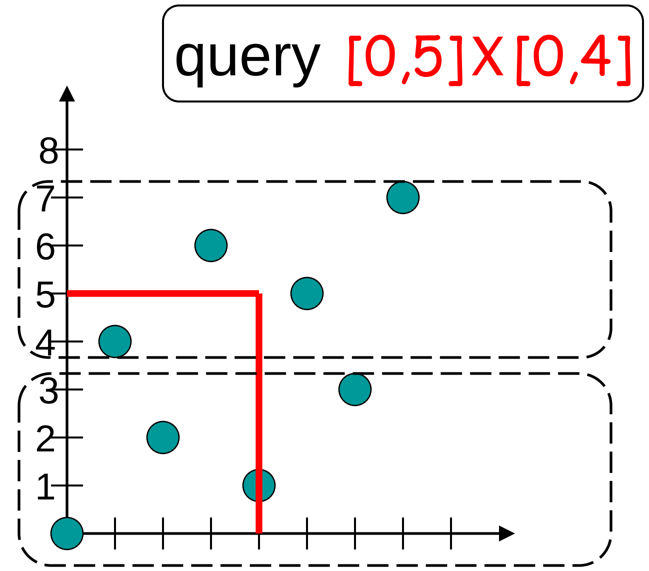




x	0	1	2	3	4	5	6	7
π(x)	0	4	2	6	1	5	3	7

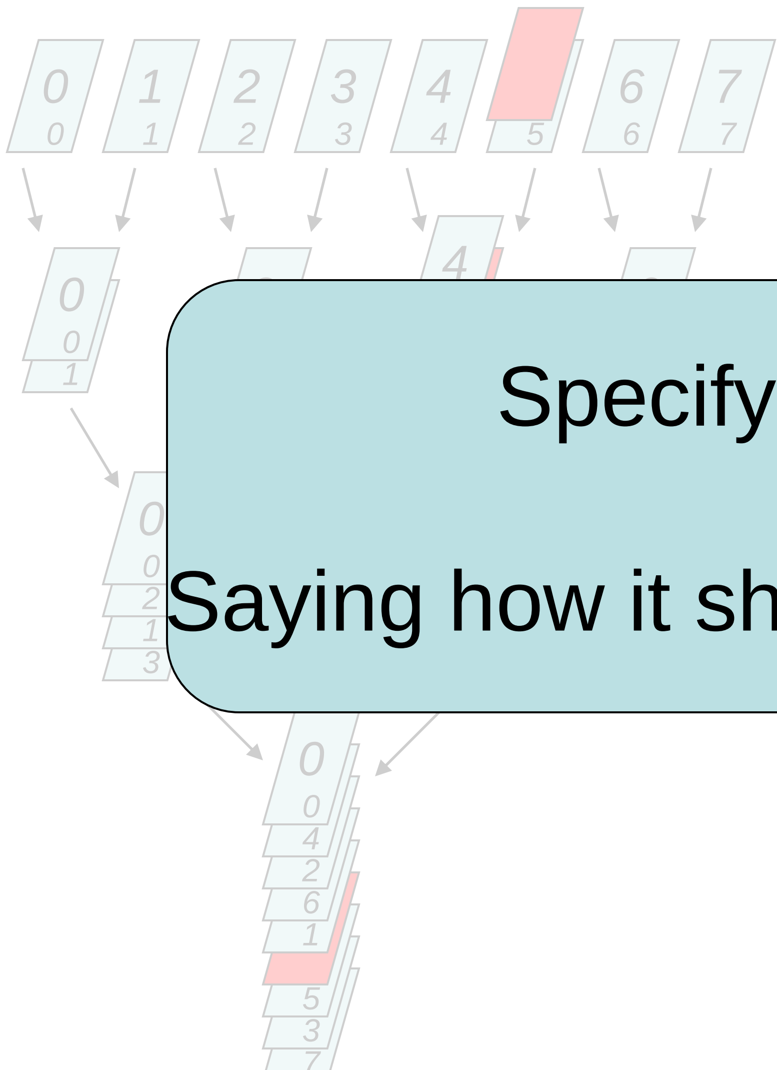
Shuffling the Queries





x	0	1	2	3	4	5	6	7
π(x)	0	4	2	6	1	5	3	7

Shuffling the Queries



Specifying a query

=

Saying how it shuffles at every level

x	0	1	2	3	4	5	6	7
π(x)	0	4	2	6	1	5	3	7

Proof Sketch (I)

- k queries, space S

  => one probe revels $\lg\binom{S}{k} = \Theta(k\lg(S/k))$

  bits of information about the queries

- t probes => $\Theta(tk\lg(S/k))$ bits

- $k=n/\lg n$ $\quad S=n\lg^{O(1)}n$ $\quad t=o(\lg n/\lg\lg n)$

  => $o(k\lg n)$ bits

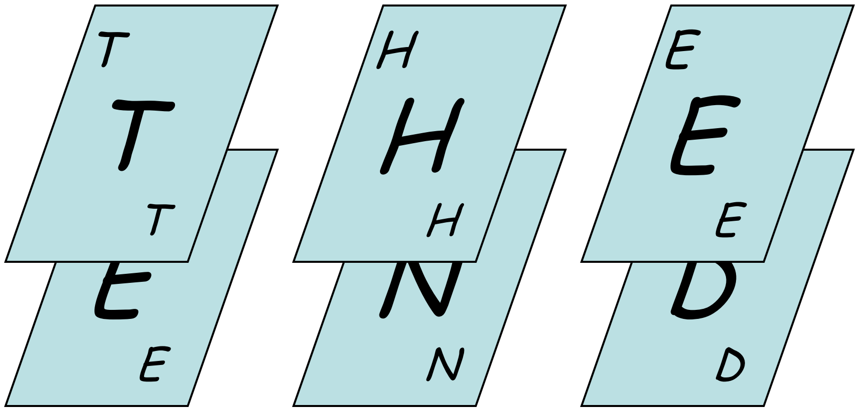
  revealed by the probes about the queries

Proof Sketch (II)

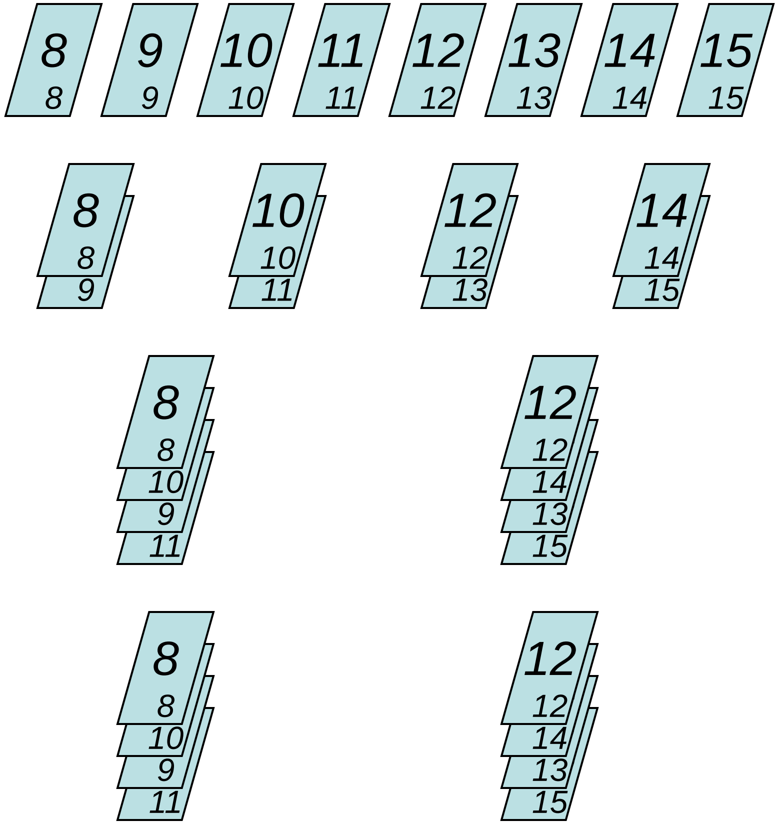
- I(queries) $\le$ I(shuffling at level 1)
  $+ I$(shuffling at level 2)
  $+ \ldots$
  $+ I$(shuffling at level $\lg n$)
  $\le o(k \lg n)$

- so for one level the data structure learns $o(k)$
  bits of information about the queries
  [i.e. doesn't know where most queries fit]

- so it cannot precompute useful sums
  [see Proceedings]



T
T
T
E
E
H
H
H
N
N
E
E
E
D
D



8
8
9
9
10
10
11
11
12
12
13
13
14
14
15
15
8
8
9
10
10
11
12
12
13
14
14
15
8
8
10
9
11
12
12
14
13
15
8
8
10
9
11
12
12
14
13
15