An introduction to Schimmler sorting

D. J. Bernstein
University of Illinois at Chicago

NSF DMS–9970409

Parallel bubble sort
on 1-dimensional mesh:
n inputs, n steps.

Schimmler sort
on 2-dimensional mesh:
n^2 inputs, $8(n-1)$ steps.

8	9	7	9	3	2	3	4
8	9	7	9	2	3	3	4
8	7	9	2	9	3	3	4
7	8	2	9	3	9	3	4
7	2	8	3	9	3	9	4
2	7	3	8	3	9	4	9
2	3	7	3	8	4	9	9
2	3	3	7	4	8	9	9
2	3	3	4	7	8	9	9

$$\frac{a \quad b}{c \quad d} \text{ means } \begin{array}{l} c = \min\{a, b\}\,, \\ d = \max\{a, b\}\,. \end{array}$$

Note that $x \mapsto [x \geq 7]$ is a min-max morphism.

"Zero-one law": $x \mapsto (r \mapsto [x \geq r])$ is injective.

1	1	1	1	0	0	0	0
1	1	1	1	0	0	0	0
1	1	1	0	1	0	0	0
1	1	0	1	0	1	0	0
1	0	1	0	1	0	1	0
0	1	0	1	0	1	0	1
0	0	1	0	1	0	1	1
0	0	0	1	0	1	1	1
0	0	0	0	1	1	1	1

kth 0 moves distance $\leq n - k$, pausing $\leq k$ times.

Any 0-1 array after n steps:

0	$\ldots$	0	1	$\ldots$	1

Any array after n steps:

$< r$	$\ldots$	$< r$	$\geq r$	$\ldots$	$\geq r$

Hence sorted.

Sorting 2-dimensional arrays

$$
\begin{array}{cccc|cccc|cccc}
0 & 1 & 1 & 2 & 2 & 1 & 1 & 0 & 0 & 1 & 1 & 2 \\
2 & 2 & 3 & 4 & 4 & 3 & 2 & 2 & 4 & 3 & 2 & 2 \\
5 & 5 & 5 & 6 & 6 & 5 & 5 & 5 & 5 & 5 & 5 & 6 \\
6 & 7 & 8 & 9 & 9 & 8 & 7 & 6 & 9 & 8 & 7 & 6 \\
\multicolumn{4}{c|}{\rightarrow} & \multicolumn{4}{c|}{\leftarrow} & \multicolumn{4}{c}{\leftrightarrow}
\end{array}
$$

Given $n \times n$ array,
n power of 2:

3	1	4	1	5	9	2	6
5	3	5	8	9	7	9	3
2	3	8	4	6	2	6	4
3	3	8	3	2	7	9	5
0	2	8	8	4	1	9	7
1	6	9	3	9	9	3	7
5	1	0	5	8	2	0	9
7	4	9	4	4	5	9	2

Recursively sort

$\rightarrow$	$\rightarrow$
$\leftarrow$	$\leftarrow$

:

1	1	2	3	2	2	2	3
3	3	3	3	4	5	5	6
3	4	4	5	6	6	7	7
5	8	8	8	9	9	9	9
1	1	0	0	2	2	1	0
4	4	3	2	5	4	4	3
7	6	5	5	9	8	7	7
9	9	8	8	9	9	9	9

Sort each column:

1	1	0	0	2	2	1	0
1	1	2	2	2	2	2	3
3	3	3	3	4	4	4	3
3	4	3	3	5	5	5	6
4	4	4	5	6	6	7	7
5	6	5	5	9	8	7	7
7	8	8	8	9	9	9	9
9	9	8	8	9	9	9	9

Sort each row, $\leftrightarrow$:

0	0	0	1	1	1	2	2
3	2	2	2	2	2	1	1
3	3	3	3	3	4	4	4
6	5	5	5	4	3	3	3
4	4	4	5	6	6	7	7
9	8	7	7	6	5	5	5
7	8	8	8	9	9	9	9
9	9	9	9	9	9	8	8

Sort each column:

0	0	0	1	1	1	1	1
3	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3
4	4	4	5	4	4	4	4
6	5	5	5	6	5	5	5
7	8	7	7	6	6	7	7
9	8	8	8	9	9	8	8
9	9	9	9	9	9	9	9

Sort each row, $\rightarrow$ (or $\leftarrow$ or $\leftrightarrow$):

0	0	0	1	1	1	1	1
2	2	2	2	2	2	2	3
3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	5
5	5	5	5	5	5	6	6
6	6	7	7	7	7	7	8
8	8	8	8	8	9	9	9
9	9	9	9	9	9	9	9

0	1	0	0	1	1	1	1
1	0	1	1	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	1	1	0	1	1
1	0	0	0	1	1	1	1
0	0	0	0	0	1	1	1
0	0	0	1	1	1	1	1
0	0	0	0	1	1	1	1

0	0	0	0	0	1	1	1
0	0	0	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
0	0	0	0	1	1	1	0
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
1	1	0	0	1	1	1	1

0	0	0	0	0	1	1	0
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
1	1	0	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

0	0	0	0	0	0	1	1
1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1
1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1
1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

Time $8(n-1)$ to sort n^2 inputs.
(Thompson-Kung 1977;
Schnorr-Shamir 1986;
Schimmler 1987)

Time $n^{2+o(1)}$ with RAM sorting.

Same hardware cost: $n^{2+o(1)}$.

Sieving by sorting

Fix nonzero $f \in \mathbf{Z}[t]$.
Given y, want to find
$\{i \in [1, y] : f(i)$ is y-smooth$\}$.

There are $y^{1+o(1)}$ pairs (q, i)
with prime q dividing $f(i)$.

Time $y^{1/2+o(1)}$ using
$y^{1/2+o(1)} \times y^{1/2+o(1)}$ mesh:
Generate pairs in parallel.
Sort in order of i.
Check y-smoothness.

Time $y^{1+o(1)}$ with RAM sieving.

Same hardware cost: $y^{1+o(1)}$.