Lecture Notes

CS377 - Parallel Programming
Marc L. Smith

Shear Sort

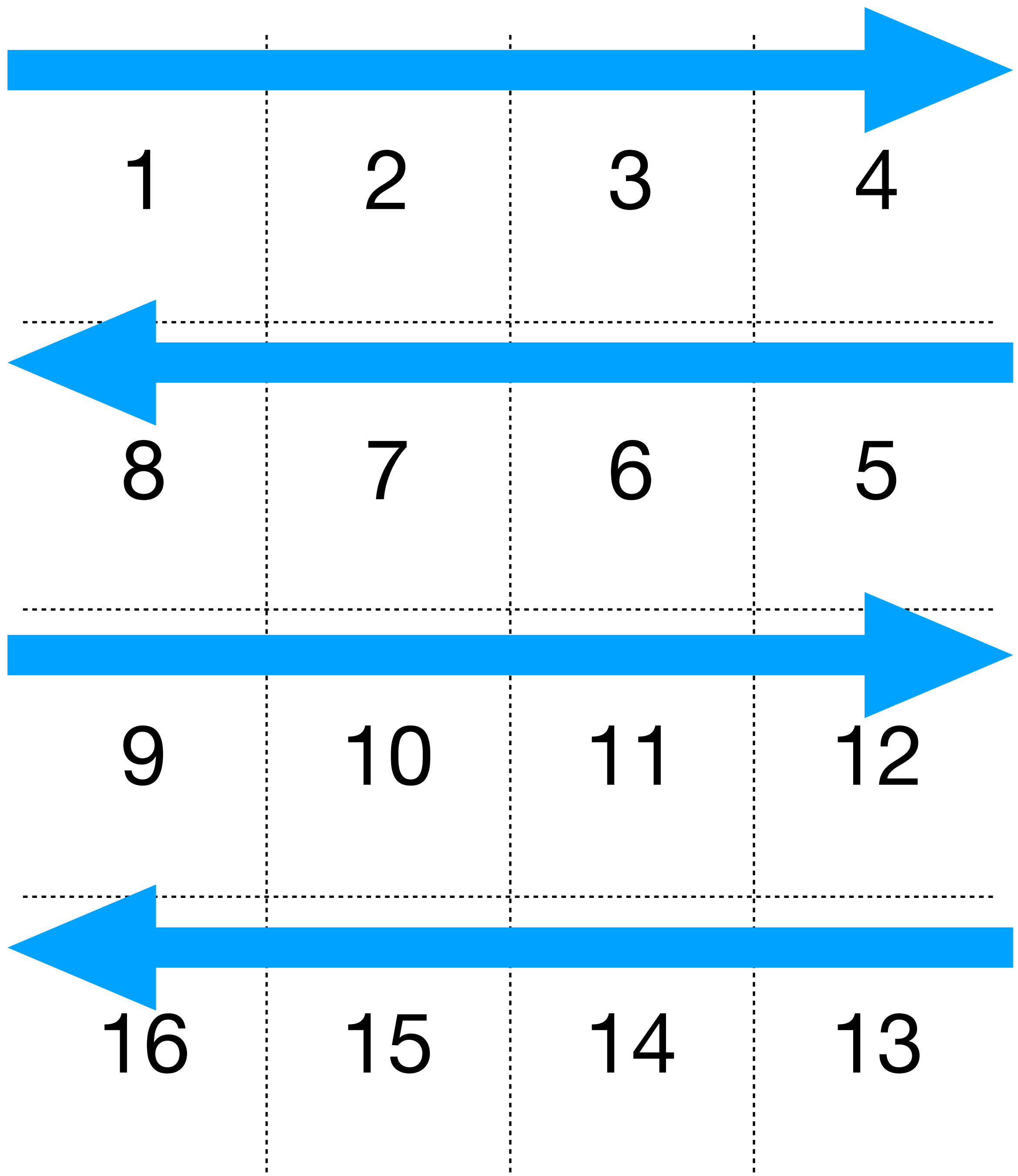
Parallel Sorting Networks

- Previously:
  - Concurrent Bubble Sort (Sort Pump Gophers)
    - One-dimensional sorting algorithm (pipeline)
    - Time Complexity: O(n)
    - Processors required: O(n)
  - Can we do better time-wise? If so, how much?
  - At what cost in number of processors required?

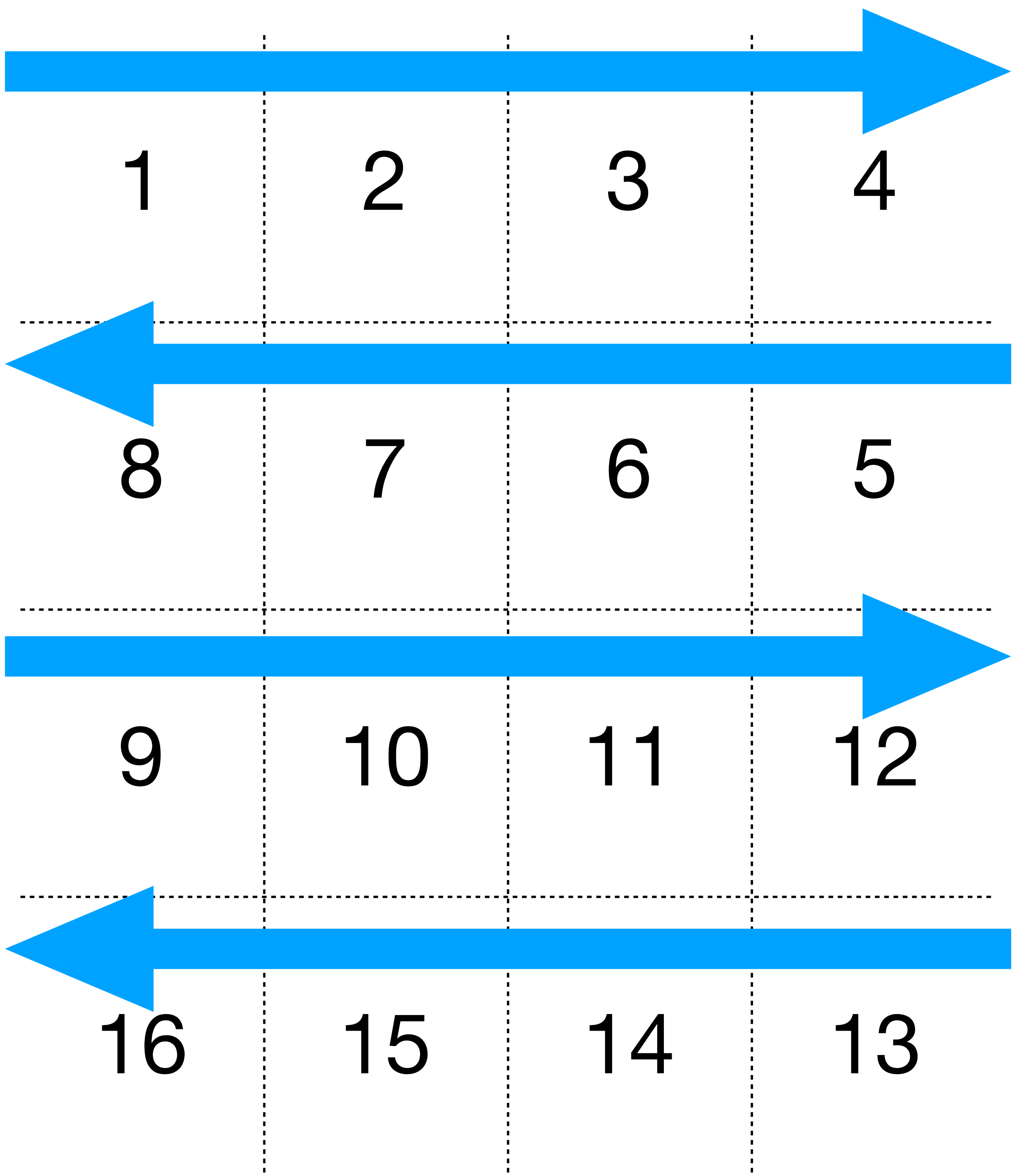
Shear Sort

- Two-dimensional sorting algorithm
- Can be implemented through either:
  - shared memory (Linda / Tuple Space — Ruby/Rinda)
  - message passing (CSP / channels — Go)
- Oblivious Comparison-Exchange (OCE) based
  - same number of comparisons regardless of initial order

Shear Sort
(snake-like order)

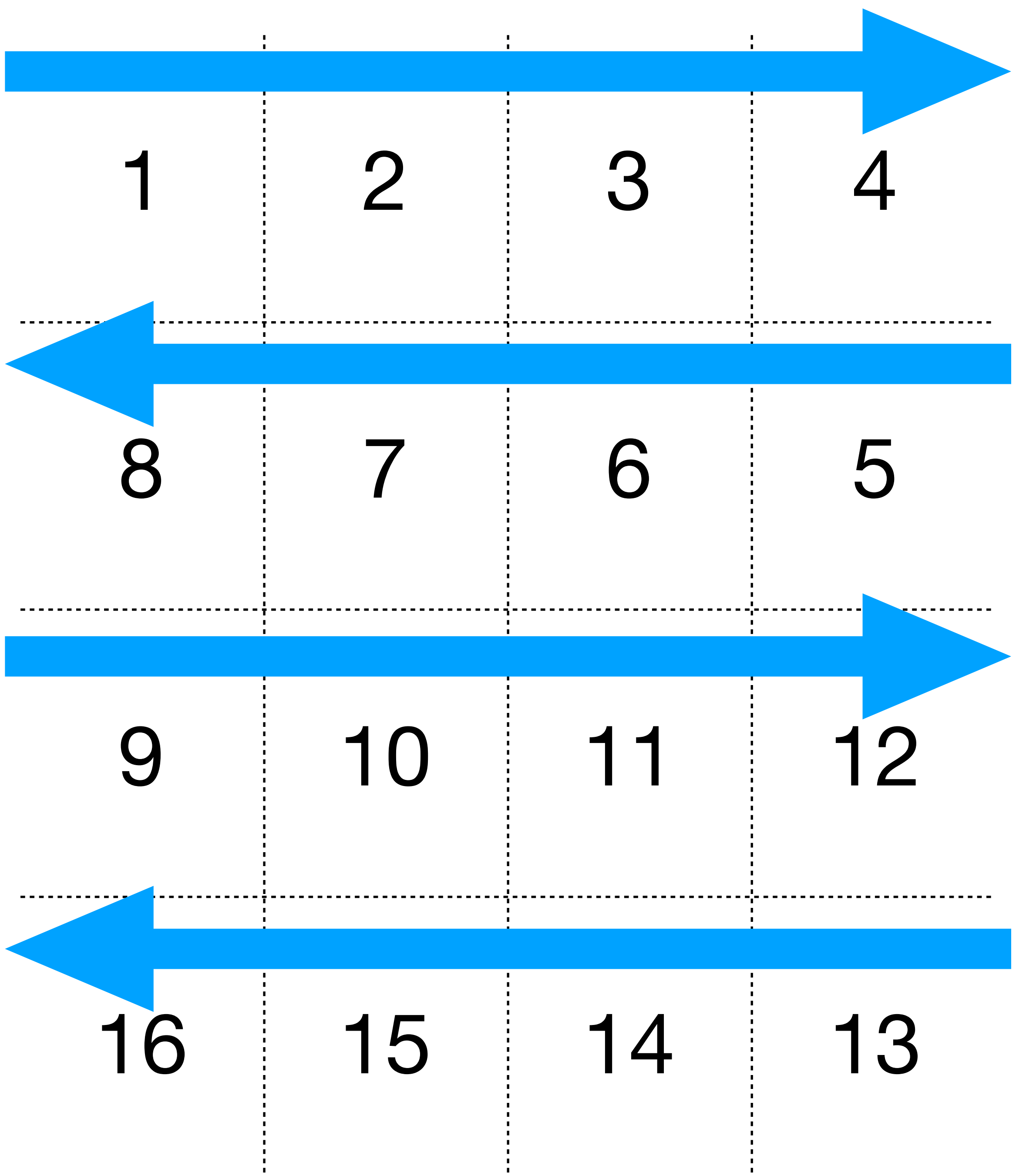


Shear Sort (implementation models)



- Linda / Tuple Space
  - each element is a tuple
- CSP
  - each element is a process
  - processes connected via channels

Shear Sort (algorithm)



- **Input:** unsorted n x n array
- **Output:** array sorted in snake-like order
- **Algorithm:**
  do log(n) times
    - sort rows (alternating order)
    - sort the columns

  sort the rows (one more time)
- **Time Complexity:** O(log(n))

Shear Sort example

3	11	6	16
8	1	5	10
14	7	12	2
4	13	9	15

Initial state

3	6	11	16
10	8	5	1
2	7	12	14
15	13	9	4

After phase 1

2	6	5	1
3	7	9	4
10	8	11	14
15	13	12	16

After phase 2

1	2	5	6
9	7	4	3
8	10	11	14
16	15	13	12

After phase 3

1	2	4	3
8	7	5	6
9	10	11	12
16	15	13	14

After phase 4

1	2	3	4
8	7	6	5
9	10	11	12
16	15	14	13

After phase 5: final

Correctness of Shear Sort

- Proof:
  - based on the 0-1 Principle (the 0-1 Sorting Lemma)
  - if algorithm works for all permutations of 0’s and 1’s, it will work for numbers of any value!
  - simplifies the number of cases we need to consider

Correctness of Shear Sort

- Assume any zero-one *n x n* matrix. There are only three kinds of rows:
  - all-one rows containing only 1's
  - all-zero rows containing only 0's
  - dirty rows containing both 0's and 1's
- Initially, input matrix can contain *n* dirty rows (worst case)
- The final matrix can contain **at most one dirty row**

Correctness of Shear Sort

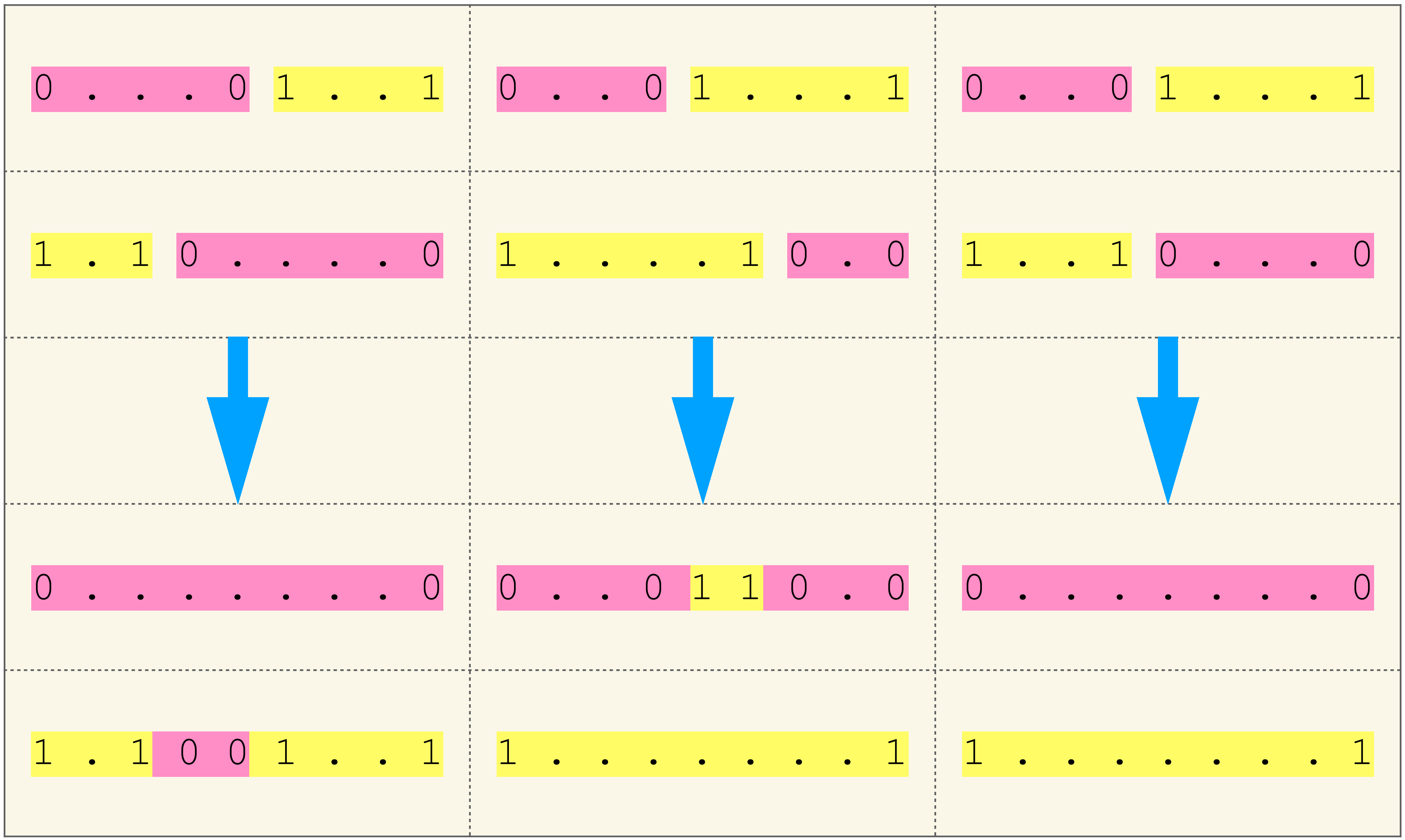
- **Proposition:** *One row and one column phase reduce the number of dirty rows to at least one half.*

- **Proof:** (case analysis)

  - Consider all dirty rows after one row phase. One half of them is sorted 0's before 1's, and the other half 1's before 0's.

  - If we consider pairs of 0-1 and 1-0 rows, we have 3 cases:

Correctness of Shear Sort
(three kinds of pairs of dirty rows)



(a) (b) (c)

Correctness of Shear Sort

- After applying one column phase:
  - one dirty row disappears in cases (a) and (b),
  - and both dirty rows disappear in case ©
- Therefore, after two log(n) phases, at most one dirty row now remains and one more row sort completes sorting
- Note: if the rows were sorted all ascending, not in snake-like order, the algorithm wouldn’t work
- Unfortunately, shear sort is not optimal. But it is cool to study!