PARALLEL PROCESSING

Mohammed Alabdulkareem

kareem@ksu.edu.sa

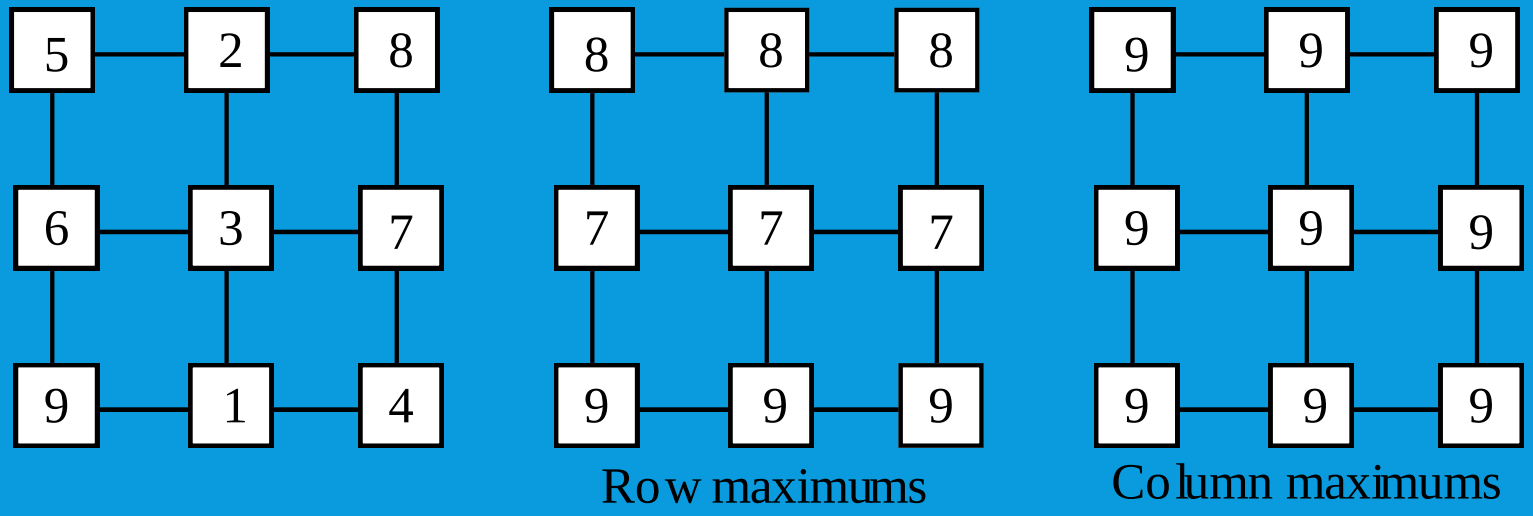
Office 2247

5

ALGORITHMS FOR 2D MESH

- **Semigroup Computation**. To perform a semigroup computation on a 2D mesh, do the semigroup computation in each row (linear Array) and then in each column.
- To find the maximum of a set of p values, stored one per processor, find the maximum of each row first and made available to every processor in the row. ($2\sqrt{p} - 2$)
- Then find column maximum for each column. ($2\sqrt{p} - 2$)
- This takes $4\sqrt{p} - 4$ steps on a p-processor square mesh.

ALGORITHMS FOR 2D MESH



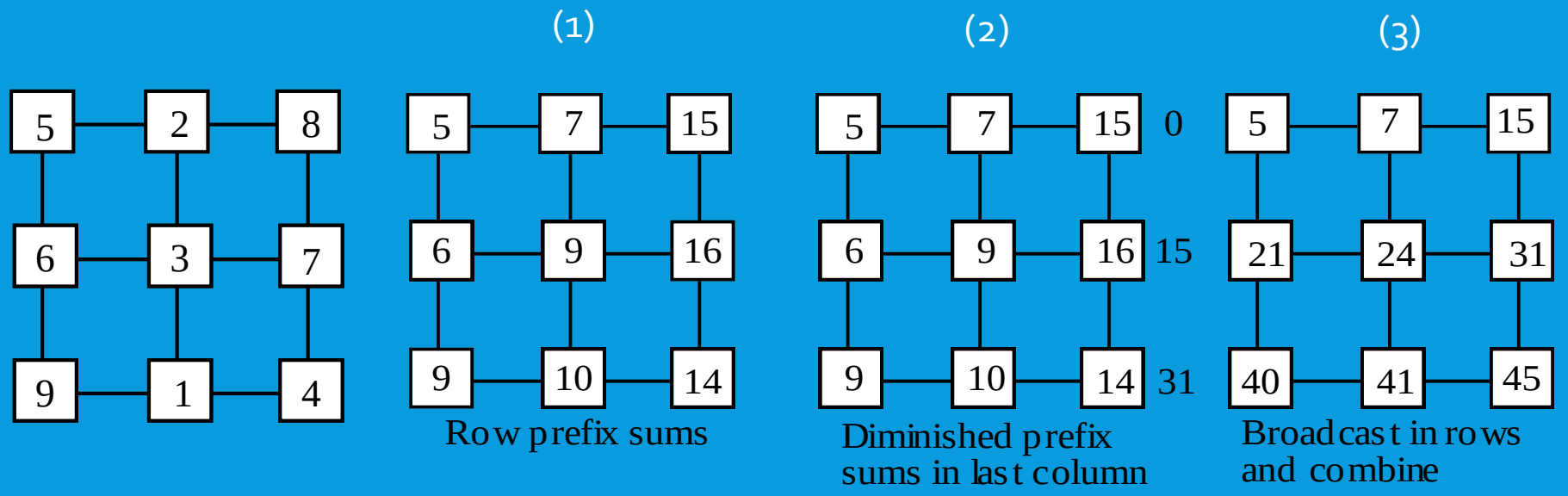
ALGORITHMS FOR 2D MESH

- **Parallel Prefix Computation**. Again, this is quite simple and can be done in three phases, assuming that the processors (and their stored values) are indexed in row-major order:
  1. do a parallel prefix computation on each row,
  2. do a diminished parallel prefix computation in the rightmost column, and
  3. broadcast the results in the rightmost column to all of the elements in the respective rows and combine with the initially computed row prefix value.

ALGORITHMS FOR 2D MESH

- For example, in doing prefix sums,
  - (1) first-row prefix sums are computed from left to right. At this point, the processors in the rightmost column hold the row sums.
  - (2) A diminished prefix computation in this last column yields the sum of all of the preceding rows in each processor.
  - (3) Combining the sum of all of the preceding rows with the row prefix sums yields the overall prefix sums.

ALGORITHMS FOR 2D MESH



ALGORITHMS FOR 2D MESH

- **Packet Routing**.
- To route a data packet from the processor in Row *r*, Column *c*, to the processor in Row *r'*, Column *c'*, we first route it within Row *r* to Column *c'*. Then, we route it in Column *c'* from Row *r* to Row *r'* . This algorithm is known as *row-first routing*. Clearly, we could do column-first routing, or use a combination of horizontal and vertical steps to get to the destination node along a shortest path.

ALGORITHMS FOR 2D MESH

Broadcasting is done in two phases:

(1) broadcast the packet to every processor in the source node's row and

(2) broadcast in all columns.

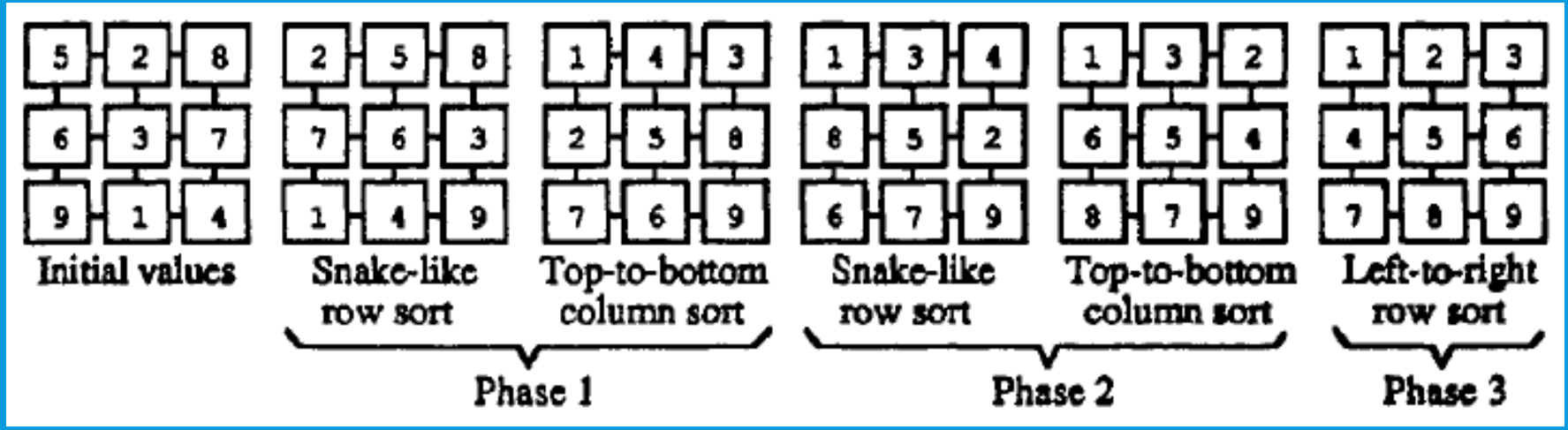
This takes at most $2\sqrt{p} - 2$ steps. If multiple values are to be broadcast by a processor, then the required data movements can be pipelined, such that each additional broadcast requires only one additional step.

ALGORITHMS FOR 2D MESH

جامعة الملك سعود
King Saud University

Shearsort

- The algorithm consists of $\lceil \log_2 r \rceil + 1$ phases in a 2D mesh with r rows.
- Row sort or column sort takes $\sqrt{p}$ compare-exchange steps.



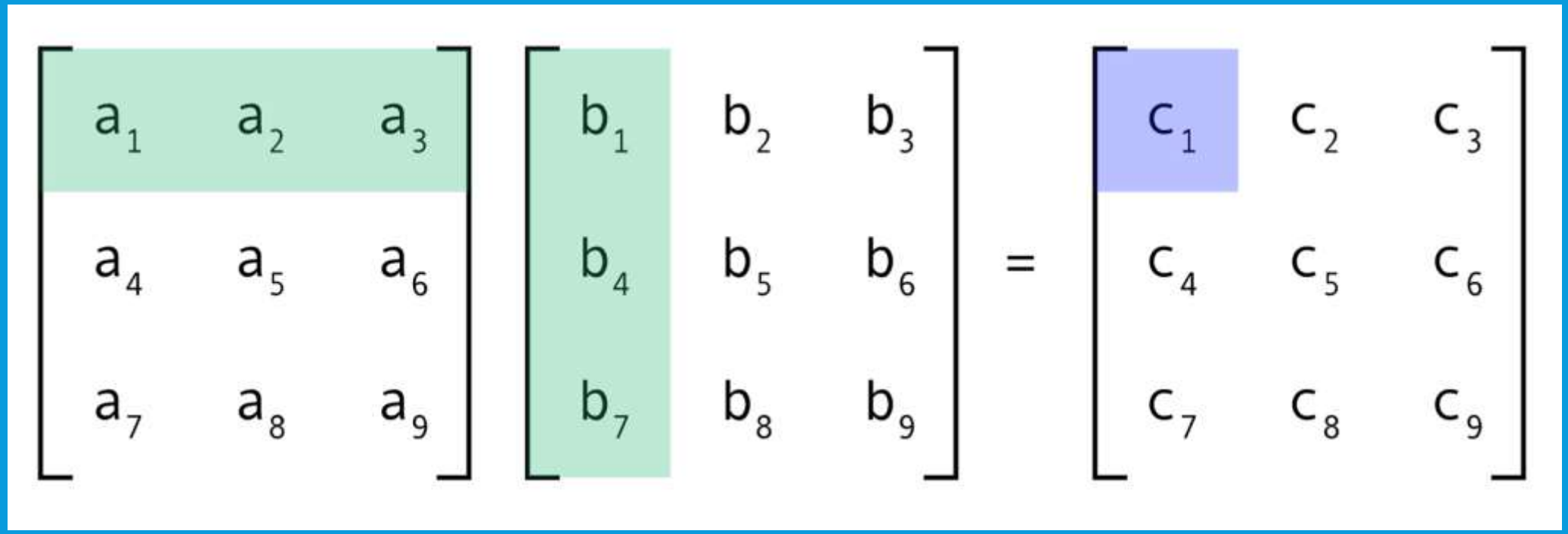
ALGORITHMS FOR SHARED VARIABLES

- Semigroup Computation.
  - Each processor obtains the data items from all other processors and performs the semigroup computation independently.
  - Obviously, all processors will end up with the same result.
- Parallel prefix
  - Each processor only obtains data items from processors with smaller indices.
- Packet routing
  - Direct communication path between any pair of processors.
- Broadcasting : each processor can send a data item to all processors directly.

ALGORITHMS FOR SHARED VARIABLES

- Sorting
  - Two phases: ranking and data permutation.
  - Ranking consists of determining the relative order of each key in the final sorted list. (*p* steps)
  - Processor *i* is responsible for ranking its own key *xi*. This is done by comparing *xi* to all other keys and counting the number of keys that are smaller than *xi*.
  - In the data permutation phase the *j*th-ranked key can be sent to Processor *j (one step)*

ALGORITHMS FOR SHARED VARIABLES



How to implement matrix multiplication on shared memory machine

ALGORITHMS FOR SHARED VARIABLES

- Assume each processor *Pi* computes one element of the result matrix.
- Processor *P1* calculates c1, P2 calculate c2, and so on.
- Processor *P1* need the row 1 from matrix A and column 1 from matrix B.

$$c1 = a1\ b1 + a2\ b4 + a3\ b7$$

- The result matrix C can be broadcasted to other processors if needed.