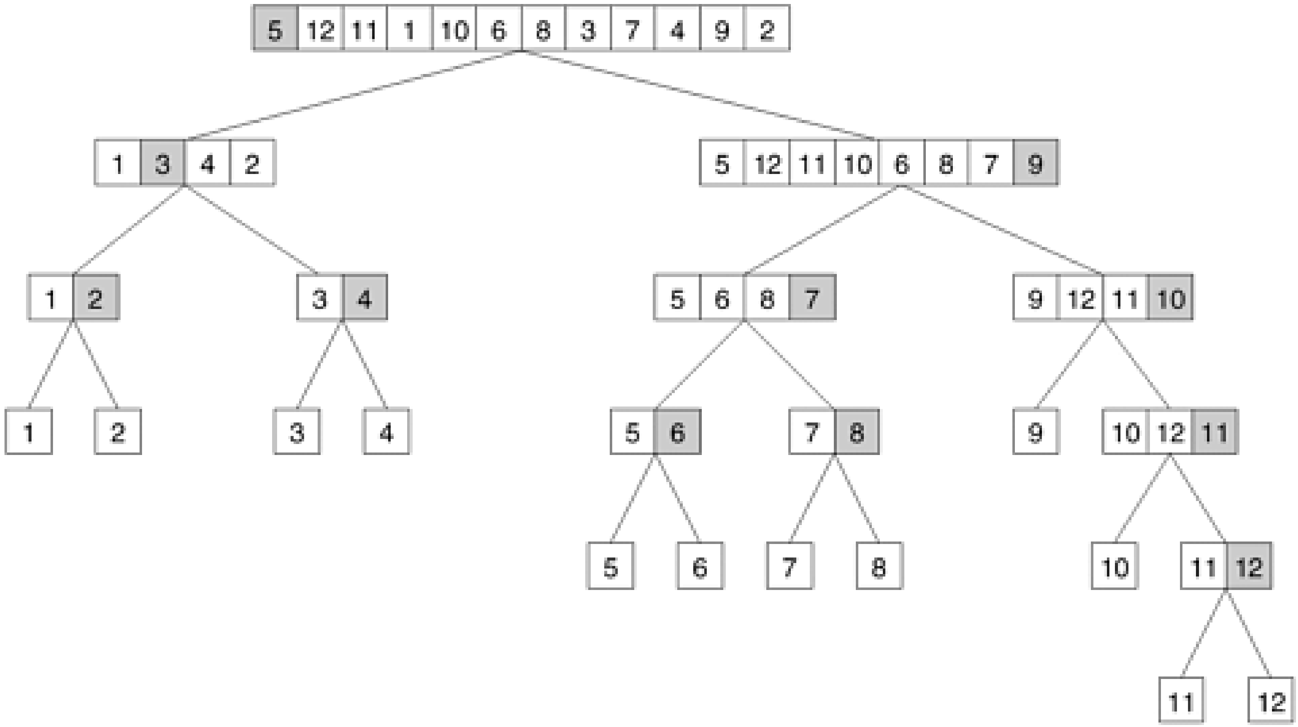
Decomposition Techniques

Decomposition techniques break down a problem into smaller tasks that can be executed in parallel. The main decomposition techniques are:

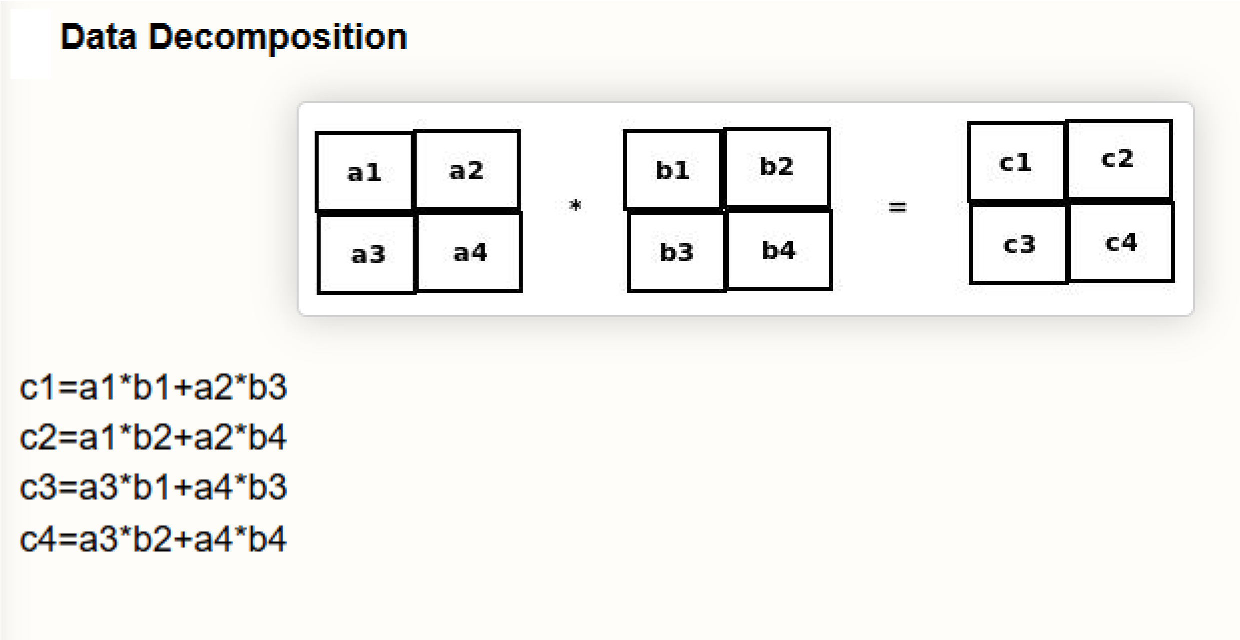
1. **Recursive Decomposition**

- The problem is divided recursively into smaller subproblems until they are simple enough to solve directly.
- **Example: Quicksort**, where the array is recursively divided into halves.



2 Data Decomposition

- Divides the data into smaller chunks, where each processing unit works on a portion of the data.
- **Types:**
    - **Block Decomposition:** Equal distribution of contiguous data blocks.
    - **Cyclic Decomposition:** Data elements are assigned in a round-robin fashion.
    - **Block-Cyclic Decomposition:** Combines block and cyclic approaches for better load balancing.
- **Example:** Parallel matrix multiplication where each thread processes different rows or columns.



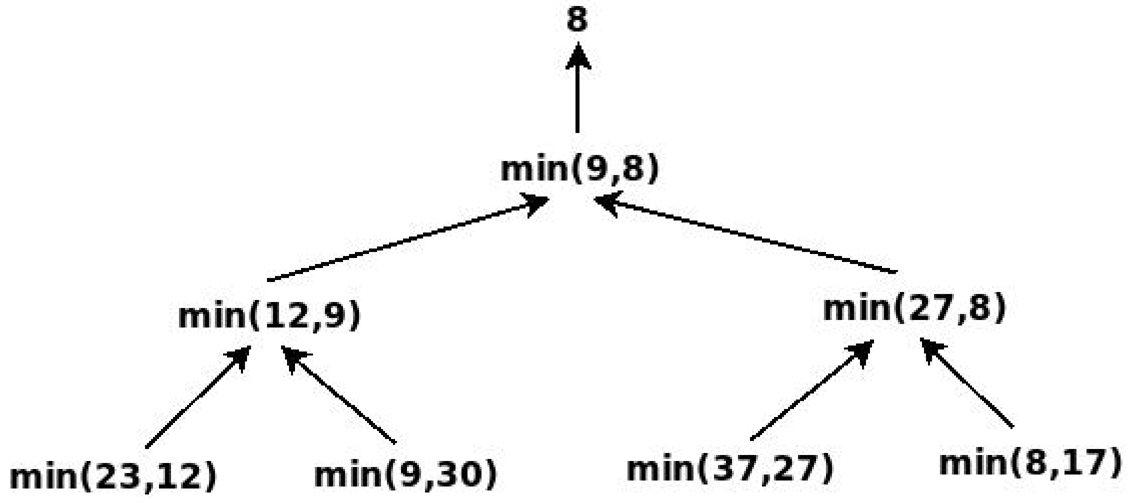
Data Decomposition

$$\begin{bmatrix} a1 & a2 \\ a3 & a4 \end{bmatrix} * \begin{bmatrix} b1 & b2 \\ b3 & b4 \end{bmatrix} = \begin{bmatrix} c1 & c2 \\ c3 & c4 \end{bmatrix}$$

c1=a1*b1+a2*b3
c2=a1*b2+a2*b4
c3=a3*b1+a4*b3
c4=a3*b2+a4*b4

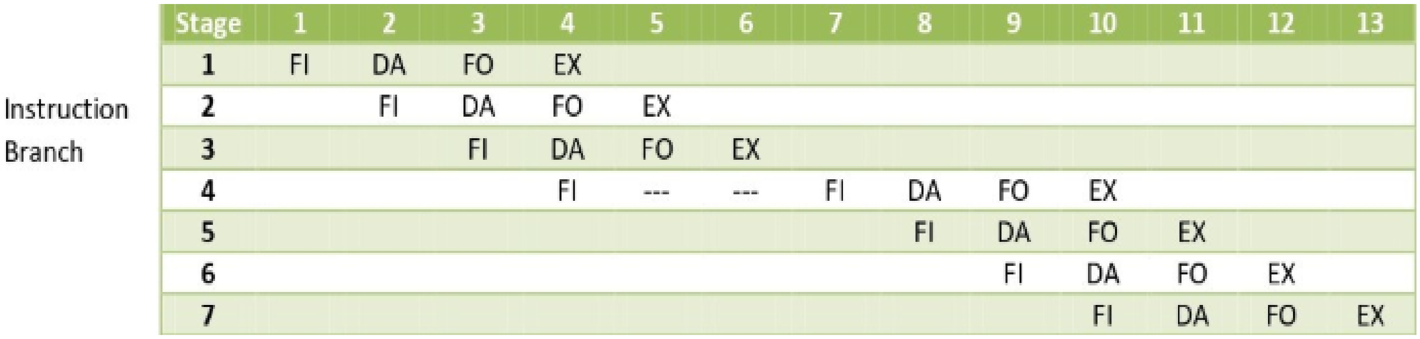
3.Task Decomposition

- Splits the tasks or computations into independent units that can execute concurrently.
- Tasks are distributed based on functionality rather than data.
- **Example:** In a web server, tasks like request handling, database queries, and response generation are parallelized.



4 Pipeline Decomposition

- The computation is divided into stages, where different processors work on different stages simultaneously.
- **Example:** Assembly line processing in image or video processing tasks.



Instruction Branch

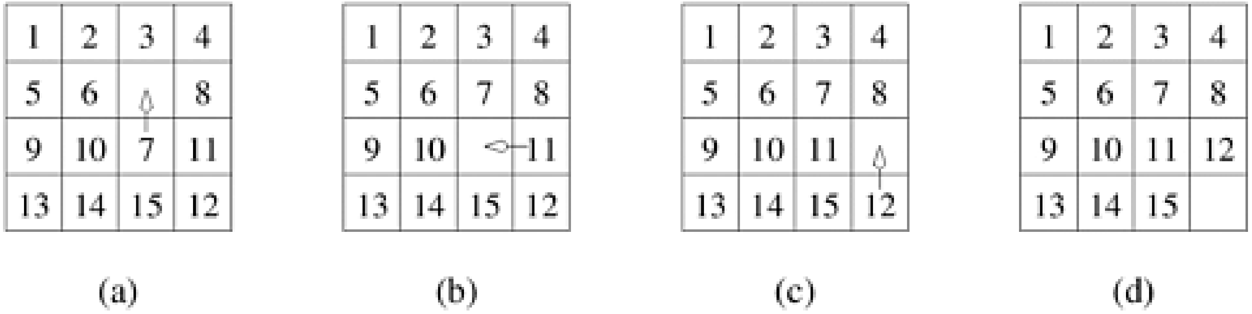
Stage	1	2	3	4	5	6	7	8	9	10	11	12	13
1	FI	DA	FO	EX									
2		FI	DA	FO	EX								
3			FI	DA	FO	EX							
4				FI	---	---	FI	DA	FO	EX			
5								FI	DA	FO	EX		
6									FI	DA	FO	EX	
7										FI	DA	FO	EX

Exploratory Decomposition

Exploratory decomposition is used to decompose problems whose underlying computations correspond to a search of a space for solutions.

In exploratory decomposition,

we partition the search space into smaller parts, and search each one of these parts concurrently, until the desired solutions are found. For an example of exploratory decomposition, consider the **15-puzzle problem**.



3. Mapping Techniques for Load Balancing

Load balancing ensures that computational tasks are evenly distributed across processors, preventing some processors from being overloaded while others remain idle.

3.1 Static Mapping

- Tasks are assigned to processors at compile time.
- Suitable for applications with predictable and uniform workloads.
- **Example:** Matrix multiplication, where matrix elements are evenly distributed.

3.2 Dynamic Mapping

- Tasks are assigned during runtime based on processor availability.
- Ideal for applications with irregular or unpredictable workloads.
- **Example:** Dynamic scheduling in task parallelism for search algorithms.

3.3 Hybrid Mapping

- Combines static and dynamic techniques to balance workload efficiently.
- **Example:** A distributed system where data is statically partitioned, but tasks are dynamically assigned.

---

4. Load Balancing Strategies

- **Work Stealing:** Idle processors steal tasks from busy ones.
- **Master-Slave Model:** A master node assigns tasks to slave nodes.
- **Graph Partitioning:** Tasks are modeled as graph nodes and balanced using graph partitioning techniques.
