PDC FINAL PREP

1. What is the primary characteristic of Distributed Shared Memory (DSM)?

   a) It utilizes a single centralized memory.

   b) It implements the shared-memory model on a physically distributed memory system.

   c) It relies solely on hardware coherence circuits.

   d) It cannot handle large databases without data replication.

2. Which method of implementing DSM involves cache coherence circuits and network interfaces?

   a) Hardware approach

   b) Page-based approach

   c) Shared variable approach

   d) Object-based approach

3. In DSM, addressing the same physical address on two processors refers to:

   a) Different locations in memory.

   b) Different memory architectures.

   c) The same location in memory.

   d) A centralized memory location.

4. Which advantage is associated with DSM?

   a) Limited scalability with a large number of nodes.

   b) Visible message passing.

   c) High cost compared to multiprocessor systems.

   d) Providing large virtual memory space.

5. What is a disadvantage of DSM compared to non-distributed shared memory systems?

a) Slower access time.

b) No need for additional protection against simultaneous accesses.

c) Greater programmer control over message generation.

d) No requirement to understand consistency models.

6. What is a common method of implementing DSM at the software level?

a) Object-based approach using fixed-size pages.

b) Page-based approach using virtual memory.

c) Tuple space organization for shared memory.

d) Hardware coherence circuits and network interfaces.

7. Which statement about software DSM systems is true?

a) They are always transparent to developers.

b) They are usually implemented as part of the underlying virtual memory architecture.

c) They provide little flexibility in organizing shared memory regions.

d) They do not require programmers to understand consistency models.

8. What is a key advantage of DSM over message-passing implementations?

a) DSM is generally cheaper.

b) DSM provides greater programmer control over messages.

c) DSM scales well with a large number of nodes.

d) DSM eliminates the need for additional protection against simultaneous accesses.

9. What type of memory architecture does a distributed-memory system consist of?

a) Single centralized memory.

b) Multiple independent processing nodes with local memory modules.

c) Hardware coherence circuits.

d) Shared variable approach using routines.

10. Which method of organizing shared memory regions involves storing shareable objects of variable sizes?

a) Page-based approach

b) Object-based approach

c) Tuple space organization

d) Shared variable approach

11. What defines a Parallel Algorithm?

a) It involves performing multiple operations simultaneously.

b) It strictly executes operations sequentially.

c) It always requires a centralized memory architecture.

d) It relies solely on hardware coherence circuits.

12. What distinguishes an inherently serial problem from others?

a) It requires extensive multiprocessing systems.

b) It can be easily divided into parallel portions.

c) It necessitates the results from a preceding step to continue.

d) It has a centralized memory architecture.

13. Which data structure is commonly used in parallel programming for its flexibility and organization?

a) Stack

b) Queue

c) Linked List

d) Binary Tree

14. What is the primary difference between statically and dynamically declared arrays?

a) Statically declared arrays have known dimensions and sizes at runtime.

b) Dynamically declared arrays have known dimensions and sizes at compile time.

c) Statically declared arrays have fixed dimensions and sizes at compile time.

d) Dynamically declared arrays have fixed dimensions and sizes at runtime.

15. Which design technique involves dividing a problem into smaller sub-problems, solving them recursively, and combining their solutions?
    a) Divide and Conquer
    b) Greedy Method
    c) Dynamic Programming
    d) Backtracking
16. In the greedy algorithm approach, what does the algorithm aim to do?
    a) It aims to solve sub-problems independently.
    b) It aims to find the globally optimized solution.
    c) It aims to find a localized optimum solution.
    d) It aims to divide the problem into smaller sub-problems.
17. Which algorithm design technique uses a brute force approach for finding the desired output by trying out all possible solutions?
    a) Divide and Conquer
    b) Dynamic Programming
    c) Backtracking
    d) Greedy Method
18. What is the primary purpose of the backtracking algorithm?
    a) To find all possible solutions for a given problem.
    b) To solve combinatorial optimization problems.
    c) To find the globally optimized solution.
    d) To divide the problem into smaller sub-problems.
19. Branch and bound is an algorithm design paradigm primarily used for solving which type of problems?
    a) Parallelizable problems
    b) Combinatorial optimization problems
    c) Inherently serial problems
    d) Dynamic programming problems
20. Which term best describes the approach of a branch and bound algorithm?
    a) It looks for the locally optimized solution.
    b) It explores all possible permutations of the solution space.

c) It strictly follows a depth-first search approach.
d) It seeks the best solution within the entire solution space.

21. Which parallel algorithm design technique involves breaking down the problem into smaller sub-problems, solving them independently, and then combining their solutions?
    a) Divide and Conquer
    b) Dynamic Programming
    c) Greedy Method
    d) Backtracking
22. What is a characteristic of dynamic programming
    a) It solves sub-problems independently.
    b) It always requires a centralized memory architecture.
    c) It stores and reuses solutions to overlapping sub-problems.
    d) It relies solely on hardware coherence circuits.
23. In the context of parallel algorithms, what does the term "backtracking" refer to?
    a) It involves solving sub-problems recursively and combining their solutions.
    b) It is an optimization technique to solve combinational problems by exploring all possible solutions.
    c) It is a process of moving backward in the solution space to explore alternative paths.
    d) It is a technique to dynamically adjust the size of the data structure based on the problem size.
24. Which design technique is commonly used for solving combinatorial optimization problems and involves merging bounds with the value of the latest best solution?
    a) Divide and Conquer
    b) Dynamic Programming
    c) Greedy Method
    d) Branch and Bound
25. What distinguishes the greedy algorithm approach from other design techniques?
    a) It aims to find the globally optimized solution.
    b) It tries to solve sub-problems recursively and combines their solutions.
    c) It selects the immediate best solution at each step without considering future consequences.

d) It divides the problem into smaller sub-problems and solves them independently.

26. Which data structure is commonly used in parallel programming for its ability to store similar types of data and support partitioning for data parallel programming?
    a) Stack
    b) Queue
    c) Linked List
    d) Array

27. Which approach involves solving sub-problems recursively and then combining their solutions to obtain the solution to the original problem?
    a) Divide and Conquer
    b) Greedy Method
    c) Backtracking
    d) Branch and Bound

28. What is the primary advantage of using a hypercube network structure in parallel algorithms?
    a) It simplifies the communication between tasks in parallel algorithms.
    b) It provides a centralized memory architecture for efficient data access.
    c) It supports dynamic adjustment of the solution space based on problem size.
    d) It enables easy embedding of other topologies such as rings and meshes.

29. In the context of parallel algorithms, which technique involves solving sub-problems recursively and using the results of overlapping sub-problems to optimize the solution?
    a) Divide and Conquer
    b) Dynamic Programming
    c) Greedy Method
    d) Backtracking

30. Which fundamental step is essential for solving a problem in parallel?
    a) Combining the computations into a single task.
    b) Dividing the computations into a set of tasks for concurrent execution.
    c) Sequentially executing all computations.
    d) Inducing recursion in the problem-solving process.

31. How are Decomposition Techniques broadly classified?
    a) As recursive and non-recursive decomposition.
    b) As sequential and concurrent decomposition.

c) As task dependency and task independence.
d) As recursive, data, exploratory, and speculative decomposition.

32. What distinguishes recursive and data decomposition techniques from speculative and exploratory decomposition techniques?
a) Recursive and data decomposition techniques apply to specific classes of problems.
b) Recursive and data decomposition techniques involve partitioning the data.
c) Speculative and exploratory decomposition techniques are more general-purpose.
d) Speculative and exploratory decomposition techniques involve concurrent search space partitioning.

33. Which decomposition technique is commonly used for problems that can be solved using the divide-and-conquer strategy?
a) Recursive decomposition
b) Data decomposition
c) Exploratory decomposition
d) Speculative decomposition

34. In data decomposition, what is the first step involved in inducing concurrency in algorithms?
a) Combining similar computations into tasks.
b) Partitioning the data on which the computations are performed.
c) Partitioning the computations into tasks.
d) Creating a task dependency graph.

35. Which decomposition technique is commonly used for algorithms that operate on large data structures?
a) Recursive decomposition
b) Exploratory decomposition
c) Speculative decomposition
d) Data decomposition

36. In data decomposition, how are computations partitioned into tasks?
a) By recursively applying division into smaller sub-problems.
b) By partitioning the data and inducing a partitioning of computations.
c) By combining the results of sub-problems.
d) By creating a task dependency graph.

37. Which type of decomposition is used for problems whose underlying computations correspond to searching a solution from a search space?

a) Recursive decomposition
b) Data decomposition
c) Exploratory decomposition
d) Speculative decomposition

38. When is speculative decomposition used?
a) When a program has a single computationally significant branch.
b) When a program may take one of many possible branches depending on the output of preceding computations.
c) When the program requires sequential execution of computations.
d) When the program involves recursive decomposition.

39. What is the scenario similar to evaluating one or more branches of a switch statement in C in parallel before the input for the switch is available?
a) Recursive decomposition
b) Data decomposition
c) Exploratory decomposition
d) Speculative decomposition

40. What is the primary goal of decomposition techniques in parallel algorithms?
a) To minimize the number of tasks for concurrent execution.
b) To maximize the size of each task for efficient computation.
c) To divide the problem into smaller tasks for concurrent execution.
d) To merge computations into a single task for sequential execution.

41. Which decomposition technique involves partitioning the search space into smaller parts and searching each part concurrently until the desired solutions are found?
a) Recursive decomposition
b) Data decomposition
c) Exploratory decomposition
d) Speculative decomposition

42. What distinguishes recursive decomposition from data decomposition?
a) Recursive decomposition involves partitioning the data, while data decomposition involves recursively applying division into smaller sub-problems.
b) Recursive decomposition involves searching a solution from a search space, while data decomposition involves combining the results of sub-problems.
c) Recursive decomposition involves inducing concurrency in problems that can be solved using the divide-and-conquer strategy, while data

decomposition involves partitioning the data on which the computations are performed.

d) Recursive decomposition is more general-purpose, while data decomposition is more specialized.

43. In data decomposition, how are computations typically partitioned into tasks?
    a) By recursively applying division into smaller sub-problems.
    b) By partitioning the data and inducing a partitioning of computations.
    c) By combining the results of sub-problems.
    d) By recursively searching for solutions in a search space.
44. Which decomposition technique is more specialized and applies to specific classes of problems?
    a) Recursive decomposition
    b) Data decomposition
    c) Exploratory decomposition
    d) Speculative decomposition
45. In speculative decomposition, what is the primary scenario where multiple tasks can start computations concurrently?
    a) When a program has a single computationally significant branch.
    b) When a program may take one of many possible branches depending on the output of preceding computations.
    c) When the program requires sequential execution of computations.
    d) When the program involves recursive decomposition.
46. What distinguishes speculative decomposition from exploratory decomposition?
    a) Speculative decomposition involves partitioning the search space, while exploratory decomposition involves recursively searching for solutions.
    b) Speculative decomposition involves evaluating one or more branches of a switch statement in C, while exploratory decomposition involves searching each part of the search space concurrently.
    c) Speculative decomposition involves searching a solution from a search space, while exploratory decomposition involves combining the results of sub-problems.
    d) Speculative decomposition involves starting computations concurrently based on the output of preceding computations, while exploratory decomposition involves partitioning the search space into smaller parts.
47. Which decomposition technique is commonly used for problems that can be solved using a divide-and-conquer strategy?

a) Data decomposition
b) Recursive decomposition
c) Exploratory decomposition
d) Speculative decomposition

48. In exploratory decomposition, what is the primary goal of partitioning the search space into smaller parts?
a) To minimize the number of tasks for concurrent execution.
b) To maximize the size of each task for efficient computation.
c) To divide the problem into smaller tasks for concurrent execution.
d) To efficiently search each part of the search space concurrently until the desired solutions are found.

49. Which decomposition technique involves partitioning the computations into tasks based on the output of preceding computations?
a) Recursive decomposition
b) Data decomposition
c) Exploratory decomposition
d) Speculative decomposition

50. What is the primary consideration when selecting the best parallelization strategy?
a) The platform's processing power
b) The size of the data structure
c) The shortest possible time for calculation
d) The application's complexity

51. Which factor determines the granularity of parallelization?
a) Number of processors available
b) Size of the data structure
c) Depth or level of details in the application
d) Type of parallelization strategy

52. What is the key characteristic of fine-grain parallelism?
a) Infrequent communication overhead
b) Small tasks with minimal code size and execution time
c) Large amounts of computation before data communication
d) Minimal load imbalance

53. What is the primary challenge associated with fine-grain parallelism?
a) High reusability of tasks
b) Poor temporal data locality
c) Low synchronization overhead

d) Reduced communication overhead

54. Which type of parallelism involves work units executing the same operations on a set of data?
    a) Data parallelism
    b) Task parallelism
    c) Pipeline parallelism
    d) Control parallelism
55. What is the primary concern of data parallelism?
    a) Minimizing load imbalance
    b) Distributing data on work units while keeping them independent
    c) Reducing communication overhead
    d) Ensuring high temporal data locality
56. What is another name for task parallelism?
    a) Functional parallelism
    b) Control parallelism
    c) Temporal parallelism
    d) Pipeline parallelism
57. How is task parallelism expressed in a multi-GPU environment?
    a) Each task is executed on a separate GPU
    b) All tasks are executed on a single GPU
    c) Tasks are executed sequentially on multiple GPUs
    d) Tasks are distributed evenly across multiple GPUs
58. What is another name for pipeline parallelism?
    a) Functional parallelism
    b) Data parallelism
    c) Temporal parallelism
    d) Control parallelism
59. What is the primary benefit of pipeline parallelism?
    a) Reduced latency and buffering
    b) Increased communication overhead
    c) Improved data replication at borders
    d) Decreased synchronization between producers and consumers
60. Which parallelization technique involves decomposing an application into a set of tasks for concurrent execution?
    a) Data parallelism
    b) Task parallelism
    c) Pipeline parallelism

d) Granularity parallelism

61. What is the primary difference between fine-grain and coarse-grain parallelism?
    a) Communication overhead
    b) Size of the data structure
    c) Depth of details in the application
    d) Degree of synchronization

62. Which type of parallelization involves dividing data into small subsets assigned to small groups of threads?
    a) Fine-grain tasking
    b) Coarse-grain tasking
    c) Data-level parallelization
    d) Task-level parallelization

63. What is the main drawback of fine-grain tiling in data-level parallelization?
    a) Reduced communication overhead
    b) High space data locality
    c) Significant overhead in processing replicated data at borders
    d) Minimal dependency management

64. Which parallelization strategy involves decomposing the program into independent sub-problems recursively?
    a) Task-level parallelization
    b) Data-level parallelization
    c) Recursive decomposition
    d) Speculative decomposition

65. What is the primary benefit of coarse-grain tasking in task-level granularity?
    a) Increased data replication at borders
    b) Improved temporal data locality
    c) Reduced synchronization overhead
    d) Minimized programming effort

66. In task parallelism, how are work units typically organized?
    a) Sequentially
    b) Independently
    c) Concurrently
    d) Hierarchically

67. What is the primary concern of pipeline parallelism?
    a) Minimizing latency
    b) Maximizing data replication
    c) Reducing synchronization

d) Balancing workload

68. Which type of parallelism involves dividing a sequence of tasks into successive phases?
    a) Task parallelism
    b) Data parallelism
    c) Pipeline parallelism
    d) Control parallelism
69. What is the key characteristic of data parallelism?
    a) Different operations on different data
    b) Same operations on different data
    c) Same operations on same data
    d) Different operations on same data
70. What is the definition of concurrency?
    a) The tendency for things to happen at the same time in any system.
    b) When multiple tasks run in overlapping periods.
    c) The illusion of multiple tasks running in parallel due to fast CPU switching.
    d) The execution of tasks in parallel across multiple CPUs.
71. Which of the following is true regarding concurrency in a single-core CPU?
    a) Two tasks can run simultaneously.
    b) Multiple tasks can run truly in parallel.
    c) Tasks run in overlapping periods.
    d) Concurrency is not possible.
72. What is the primary difference between concurrency and parallelism?
    a) Concurrency is the execution of tasks in parallel, while parallelism is the tendency for things to happen simultaneously.
    b) Concurrency involves multiple tasks running in overlapping periods, while parallelism involves tasks actually running in parallel in multiple CPUs.
    c) Concurrency relies on multiple processors, cores, or threads, while parallelism is an illusion created by fast CPU switching.
    d) Concurrency is the execution of tasks in multiple CPUs, while parallelism is the execution of tasks in a single CPU.
73. Which aspect is important when dealing with concurrency issues in software systems?
    a) Detecting and responding to external events occurring in a random order.

b) Ensuring that concurrent activities evolve independently.
c) Creating separate programs for each concurrent activity.
d) Achieving maximum parallelism.

74. What is synchronization in the context of concurrent tasks?
a) The execution of tasks in multiple CPUs.
b) The coordination of concurrent tasks to prevent interference or unsafe access to shared resources.
c) Ensuring that tasks run in overlapping periods.
d) Achieving maximum parallelism.

75. Which of the following is a potential risk introduced by concurrency and synchronization?
a) Deadlocks
b) Memory leaks
c) Race conditions
d) All of the above

76. What is the purpose of mutual exclusion in concurrency control?
a) To allow multiple threads to enter a critical section simultaneously.
b) To prevent race conditions by ensuring only one thread enters a critical section at a time.
c) To maximize parallelism in the system.
d) To synchronize multiple CPUs.

77. What is a race condition in the context of concurrent tasks?
a) A condition where multiple threads enter a critical section simultaneously.
b) A condition where shared resources are accessed consistently.
c) An undesirable event occurring when multiple entities access or modify shared resources inconsistently.
d) A condition where multiple CPUs synchronize their operations.

78. What is a semaphore in the context of concurrency control?
a) A mechanism that enforces limits on accessing a resource.
b) An integer variable supporting atomic operations for controlling access to shared resources.
c) A datatype proposed by Edsger Dijkstra to prevent race conditions.
d) A variable that changes depending on conditions.

79. What are the two atomic operations supported by a semaphore?
a) Pass and Release
b) Increment and Decrement

c) P and V
d) Lock and Unlock