Quiz: Module 3 428 - MapReduce Programming

1. What is the fundamental principle behind MapReduce?

a) Divide and conquer

b) Brute force

c) Recursion

d) Dynamic programming

2. Which function in MapReduce is responsible for aggregating intermediate data?

a) Map function

b) Reduce function

c) Combiner function

d) Partitioner function

3. Which of the following is a classic example demonstrating the core of MapReduce?

a) Inverted Index creation

b) Word Count program

c) Matrix Multiplication

d) Log Analysis

4. In the context of MapReduce, what is an inverted index primarily used for?

a) Efficient sorting of large datasets

b) Fast search in information retrieval systems

c) Distributed matrix multiplication

d) Web server log analysis

5. Which technique helps in reducing the amount of data transferred between map and reduce phases in MapReduce?

a) Partitioner

b) Combiner

c) Speculative Execution

d) Data Locality

6. What is the purpose of a Partitioner in MapReduce?

a) Reduce the size of intermediate data

b) Optimize data distribution among reducers

c) Handle slow nodes in the cluster

d) Filter out unwanted data

7. Which design pattern in MapReduce is suitable for calculating the average value of data across distributed nodes?

a) Counting

b) Averaging

c) Top-K Filtering

d) Bloom Filtering

8. Which filtering pattern in MapReduce uses a probabilistic method to filter large sets with low memory usage?

a) Counting

b) Averaging

c) Top-K Filtering

d) Bloom Filtering

9. Which join pattern in MapReduce performs joins in the mapper to reduce network overhead?

a) Reduce-Side Join

b) Map-Side Join

c) Secondary Sorting

d) Total Sort

10. What is the main advantage of using Secondary Sorting in MapReduce?

a) Reduce network overhead

b) Sort records by both key and an additional field

c) Ensure global sorting of output data

d) Filter large datasets efficiently

11. In which phase of MapReduce job execution is the input data divided into fixed-size chunks?

a) Input Split

b) Map Execution

c) Shuffle and Sort

d) Reduce Execution

12. What is the role of the Shuffle and Sort phase in MapReduce?

a) Divide input data into chunks

b) Process data in parallel using map tasks

c) Group data with the same key

d) Generate final results

13. Which of the following is NOT a technique used for optimizing MapReduce job performance?

a) Combiner

b) Partitioner

c) Bloom Filtering

d) Data Locality

14. Which MapReduce application development step involves defining how input data is split and output data is written?

a) Problem Definition

b) Mapper Implementation

c) Reducer Implementation

d) Input/Output Format

15. Which MapReduce application is suitable for analyzing web server logs to extract useful information?

a) Word Count

b) Inverted Index creation

c) Log Analysis

d) Recommendation Systems

16. What is the primary purpose of using MapReduce for Recommendation Systems?

a) Analyze web server logs

b) Calculate similarity scores between users or items

c) Search for specific patterns in text files

d) Multiply large matrices

17. Which technique is used in MapReduce to handle situations where certain keys in the data are much more frequent than others?

a) Data Skew Handling

b) Compression

c) Speculative Execution

d) Data Locality

18. What is the benefit of using Compression in MapReduce?

a) Handle imbalanced workload across reducers

b) Reduce the size of intermediate data

c) Run duplicate tasks to avoid bottlenecks

d) Schedule tasks on nodes where data is stored

19. Which MapReduce design pattern is appropriate for calculating statistics like mean, median, and standard deviation for a large dataset?

a) Numerical Summarization

b) Graph Algorithms

c) Reduce-Side Join

d) Map-Side Join

20. Which MapReduce design pattern is used to implement graph algorithms like PageRank in a distributed manner?

a) Numerical Summarization

b) Graph Algorithms

c) Reduce-Side Join

d) Map-Side Join

Answers:

1. a
2. b
3. b
4. b
5. b
6. b
7. b
8. d
9. b
10. b
11. a
12. c
13. c
14. d
15. c
16. b
17. a
18. b
19. a
20. b