4. Match the following sets and relations with their properties, considering their mathematical definitions:
1. Properties of Relations Characteristic
I. Symmetric A. For any (a, b), if (a, b) ? R then (b, a) ? R
II. Transitive B. For any (a, b) and (b, c), (a, c) ? R
III. Equivalence Relation C. A relation that is reflexive, symmetric, and transitive
IV. Partial Order D. A relation that is reflexive, antisymmetric, and transitive
Choose the correct answer from the options given below:
(1) I-C, II-D, III-A, IV-B
(2) I-A, II-B, III-C, IV-D
(3) I-D, II-A, III-B, IV-C
(4) I-B, II-C, III-D, IV-A
Answer Key: 2
Solution:
? Symmetric property (I) states that if (a, b) ? R then (b, a) ? R, matching option A.
? Transitivity (II) ensures that if (a, b) and (b, c) are in R, then (a, c) must also be in R, matching B.
? An Equivalence Relation (III) is characterized by being reflexive, symmetric, and transitive, thus matching C.
? A Partial Order (IV) is a relation that is reflexive, antisymmetric, and transitive, aligning with D.
Hence, Option (2) is the right answer.

Match the following syntax analysis techniques with their defining characteristics, considering their approach to parsing:
1. Associativity and Precedence Characteristic
I. Recursive Descent A. Primarily top-down, uses lookahead and predictive parsing
II. LL(1) Parsing B. Uses a stack, performs left-to-right parsing with one lookahead token
III. LR Parser C. Handles more complex grammars via shift-reduce methodology
IV. Bottom-up Parsing D. Deals with associativity and precedence via grammar transformations
Choose the correct answer from the options given below:
(1) I-D, II-A, III-C, IV-B
(2) I-A, II-B, III-D, IV-C
(3) I-B, II-D, III-A, IV-C
(4) I-C, II-B, III-D, IV-A
Answer Key: 4
Solution:
? Associativity and precedence influence grammar transformations and parsing strategies, especially in bottom-up parsing methods.
? Recursive Descent and LL(1) are top-down parsers, relying on lookahead and grammar predictions, less directly concerned with associativity.
? LR parsers, a class of bottom-up parsers, manage shift-reduce operations that inherently handle associativity and precedence via grammar rules.
? Bottom-up parsing techniques systematically reduce input to start symbols, naturally integrating precedence, and associativity considerations.
Hence, Option (4) is the right answer.

Match the following Activation Record components with their roles in runtime storage organization:
1. Activation Record Component Role
A. Return Address 1. Stores the address to return to after function call
B. Local Variables 2. Holds temporary data and function parameters
C. Control Link 3. Maintains control flow context
D. Dynamic Link 4. Links to the caller's activation record for nested calls
Choose the correct answer from the options given below:
(1) A-1, B-2, C-3, D-4
(2) A-2, B-1, C-4, D-3

(3) A-3, B-4, C-1, D-2
(4) A-4, B-3, C-2, D-1
Answer Key: 2
Solution:
? Return Address: Stored to know where to return after function execution, linked with activation record's control flow.
? Local Variables: Stored within activation record for each function invocation to preserve data specific to that call.
? Control Link: Points to the previous activation record, establishing the call stack structure.
? Dynamic Link: Connects to the caller's activation record, enabling access to non-local data and maintaining call hierarchy.
Hence, Option (2) correctly matches the components with their roles.

Match the following distributed system characteristics with their implications:
1. Robustness in Distributed File Systems Implication
A. Ensures data availability despite node failures
B. Requires redundancy and replication
C. Increases complexity in synchronization
D. Guarantees data consistency across nodes
Choose the correct answer from the options given below:
(1) A-B, B-C, C-D, D-A
(2) A-C, B-A, C-D, D-B
(3) A-B, B-D, C-C, D-A
(4) A-D, B-C, C-B, D-A
Answer Key: 3
Solution:
? Robustness ensures data availability even when some nodes fail, necessitating redundancy and replication.
? Replication introduces complexity in maintaining consistency and synchronization.
? Synchronization mechanisms are needed to ensure data consistency across distributed nodes.
? Data consistency guarantees are critical but require careful coordination, often increasing system complexity.
Hence, Option (3) correctly reflects the implications of robustness in distributed file systems.

3. Match the following data processing models with their key characteristics, reflecting their handling of distributed data and fault tolerance:
1. Big Data Architecture Characteristic
I. MapReduce A. Processes large datasets by mapping and reducing, inherently fault-tolerant
II. Hadoop Distributed File System B. Provides distributed storage with replication for fault tolerance
III. Data Lake C. Centralized storage for raw data, schema on read approach
IV. Distributed File System D. Supports scalable, distributed data processing frameworks
Choose the correct answer from the options given below:
(1) I-D, II-B, III-C, IV-A
(2) I-A, II-D, III-B, IV-C
(3) I-B, II-A, III-D, IV-C
(4) I-C, II-B, III-A, IV-D
Answer Key: 1
Solution:
? MapReduce leverages a model where data is processed in distributed tasks with inherent fault tolerance via task re-execution, fitting characteristic A.
? Hadoop Distributed File System (HDFS) offers distributed storage with data replication, ensuring fault tolerance, matching characteristic B.
? Data Lake is a centralized repository storing raw data with a schema-on-read approach, aligning with characteristic C.

? Distributed File Systems underlie frameworks like HDFS, providing scalable, distributed storage, matching characteristic D.
Hence, Option (1) is the right answer.

3. Match the following graph algorithms with their primary operational characteristics:
1. Breadth-First Search (BFS) Characteristic
A. Explores all neighbors before moving deeper
B. Finds the shortest path in unweighted graphs
C. Uses a queue data structure for traversal
D. Suitable for connectivity and level-wise traversal
Choose the correct answer from the options given below:
(1) A-B, B-C, C-D, D-A
(2) A-C, B-A, C-D, D-B
(3) A-D, B-B, C-A, D-C
(4) A-B, B-D, C-A, D-C
Answer Key: 2
Solution:
? BFS explores nodes level by level, which corresponds to exploring all neighbors before deeper nodes, using a queue.
? It finds shortest paths in unweighted graphs due to uniform traversal depth.
? The queue ensures the order of exploration, maintaining the level-wise approach.
? It?s widely used for connectivity checks and level-based traversal.
Hence, Option (2) correctly pairs the algorithm with its characteristic.