From: Bl4z3
Module 1

1. What is an algorithm?
   - A) A type of computer software.
   - B) A step-by-step method for solving a well-defined problem.
   - C) A programming language.
   - D) A mathematical equation.

2. Which of the following is NOT a characteristic of an algorithm?
   - A) It takes acceptable inputs.
   - B) It produces a desired output.
   - C) It contains ambiguous instructions.
   - D) It consists of a set of instructions.

3. Where does the term "algorithm" originate from?
   - A) A Greek mathematician.
   - B) The name of a Persian mathematician, Muhammad ibn Musa al-Khwarizmi.
   - C) The Latin word for "problem."
   - D) A type of ancient computer.

4. Which of the following best describes the importance of an algorithm being unambiguous?
   - A) It can be interpreted in multiple ways.
   - B) It must be understandable to a human.
   - C) It ensures that the problem is stated clearly.
   - D) It makes the algorithm faster.

5. What is the main purpose of an algorithm?
   - A) To execute a computer program.
   - B) To solve a problem with a defined set of instructions.
   - C) To store data in a computer.
   - D) To translate a programming language.

6. Who is Muhammad ibn Musa al-Khwarizmi, and what is his significance in mathematics?
   - A) The inventor of the first computer.
   - B) A mathematician whose works introduced Arabic numerals and algebraic concepts to Western mathematics.
   - C) The creator of the binary system.
   - D) A philosopher who wrote about logic.

7. The word "algebra" originates from which book written by al-Khwarizmi?
   - A) "The Computation Book"
   - B) "The Book of Numbers"
   - C) "Kitab al-Jabr wa'l-Muqabala"
   - D) "The Book of Algorithms"

8. Why is it important for the instructions in an algorithm to be explicitly stated?
   - A) To allow human intuition to play a role.
   - B) To ensure the problem solver understands the solution process.
   - C) To make the algorithm more complex.
   - D) To reduce the number of steps in the algorithm.

9. What would happen if an algorithm receives incorrect input?
   - A) It will automatically correct the input.
   - B) It will produce an incorrect output.
   - C) It will not execute.
   - D) It will produce the correct output regardless of the input.

10. Which of the following is an example of an algorithm in daily life?
   - A) A computer's motherboard.
   - B) A cooking recipe booklet.
   - C) A song playlist.
   - D) A movie's script.

11. Consider the problem: "Find the largest of three numbers." Which of the following is the correct step if `a > b`?
   - A) Compare `b` and `c`.
   - B) Set `b` as the largest.
   - C) Compare `a` and `c`.
   - D) Set `a` as the smallest.

12. Which of the following steps is NOT included in the algorithm to find the mean of a set of numbers?
   - A) Initialize `sum` to 0.
   - B) Initialize `i` to 0.
   - C) Subtract each number from the `sum`.
   - D) Divide the `sum` by `n`.

13. Which of the following is NOT an example of an algorithm?
   - A) The rules of how to play a game.
   - B) Instructions for assembling furniture.
   - C) A random selection of songs.
   - D) Directional maps for driving from point A to B.

14. Which of the following best defines problem-solving?
   - A) Guessing the solution to a problem.
   - B) The act of finding a solution to a difficulty or constraint.
   - C) A natural ability to solve issues without any process.
   - D) Using random methods to tackle challenges.

15. Which of the following is NOT considered a systematically effective method for solving problems?
- A) Experience
- B) Process model
- C) Trial and error
- D) Using algorithms

16. What is the main focus when discussing algorithms in problem solving?
- A) Guesswork and luck
- B) Systematically effective methods
- C) Trial-and-error methods
- D) Natural endowment

17. In computer science, what is a crucial element in solving problems with computers?
- A) Guessing the right algorithm
- B) Understanding the computer's information processing model
- C) Using random inputs and outputs
- D) Depending solely on natural problem-solving abilities

18. What does the process model represent in problem-solving?
- A) A set of instructions based on someone else's experience.
- B) Scientific steps that systematically solve a problem.
- C) A random approach to tackling problems.
- D) A method based on guesswork.

19. Which component of the computer's information processing model involves obtaining data from the user?
- A) Processing
- B) Output
- C) Input
- D) Storage

20. Which device is used for processing inputs in a computer?
- A) Display unit
- B) Central Processing Unit (CPU)
- C) Keyboard
- D) Printer

21. How are problems solved using a computer according to the information processing model?
- A) By using random guesses to find the solution.
- B) By inputting data, processing it, and producing output data.
- C) By relying on trial and error.
- D) By using a natural endowment approach.

22. Which of the following is NOT an output device in a computer system?
- A) Speaker
- B) Display unit
- C) Storage drive
- D) Printer

23. Which of the following best describes data transmission in a computer's information processing model?
- A) The CPU processes inputs to produce data.
- B) Input/output data may be transmitted via storage drives and network devices.
- C) The user inputs data using devices like a keyboard or mouse.
- D) Output data is produced by devices such as speakers and printers.

24. What does the computer model of computation rely on?
- A) Guesswork and luck.
- B) Input, processing, and output.
- C) Random inputs.
- D) Instinctive problem-solving.

25. In the context of problem-solving, what is the first step to take?
- A) Formulating a model.
- B) Guessing the solution.
- C) Understanding the problem.
- D) Computing the output directly.

26. Which of the following is crucial when understanding a problem?
- A) Guessing the format of the data.
- B) Understanding the format and nature of input data.
- C) Directly outputting the data.
- D) Ignoring any missing data.

27. Why is it important to know the format of the input data when solving a problem?
- A) To guess the output format.
- B) Because different formats require different processing methods.
- C) To ignore the incorrect data.
- D) Because the format does not affect the processing.

28. If grades are represented as letters (e.g., A, B, C), what should be done to calculate an average?
- A) Add and divide the letters directly.
- B) Assign numerical values to the letters, then calculate.
- C) Ignore the letter grades.
- D) Guess the average.

29. What is the result of the computational model if the input is a bunch of numerical grades?
- A) A random number.
- B) A number from 0 to 100.
- C) A letter grade.
- D) A pie chart.

30. What should be considered if some grades are missing in a dataset?
- A) Include those students with a grade of 0.
- B) Ignore the problem.
- C) Calculate the average without considering missing grades.
- D) Assign the highest grade to missing entries.

31. When assigning numbers to letter grades (e.g., A = 12, B = 11), what is the purpose?
- A) To complicate the process.
- B) To make it possible to perform arithmetic operations like addition and division.
- C) To guess the average.
- D) To make the data harder to process.

32. What is the purpose of a "lookup table" in the context of computing an average from letter grades?
- A) To randomly generate letter grades
- B) To map numeric averages back to corresponding letter grades
- C) To store large amounts of data efficiently
- D) To visually represent data

33. What is the main goal of the second step, "Formulating a Model," in problem-solving?
- A) To figure out how to use the available data to compute an answer
- B) To create a flowchart
- C) To implement the algorithm
- D) To determine the final output format

34. Which of the following is a commonly used representation of an algorithm?
- A) Data table
- B) Pseudo-code
- C) Pie chart
- D) File format

35. What is pseudo-code primarily used for?
- A) To describe an algorithm in a way that is understandable to humans
- B) To execute a program on a computer
- C) To visualize data
- D) To store data in a file

36. In the "DisplayGrades" algorithm, what is the primary purpose of the loop (repeated n times)?
- A) To display each grade individually
- B) To sum all the grades
- C) To find the highest grade
- D) To calculate the percentage of each grade

37. What is the result of dividing the sum of grades by the number of grades in the "DisplayGrades" algorithm?
- A) The total number of grades
- B) The average grade
- C) The maximum grade
- D) The minimum grade

38. In problem-solving, when should you quit the process according to the lamp problem example?
- A) When the lamp works
- B) When the bulb is replaced
- C) When the lamp is plugged in
- D) When a new lamp is purchased

39. In the lamp problem example, what should you do if the lamp is not working after replacing the bulb?
- A) Buy a new lamp
- B) Replace the socket
- C) Check the wiring
- D) Contact customer support

40. What is the main purpose of running a program after writing and compiling it?
- A) To check the file format
- B) To ensure it solves the intended problem and produces correct solutions
- C) To generate a random output
- D) To display the code on the screen

41. If a program runs correctly for some input data but not for all, what might be the issue?
- A) The algorithm might not handle all possible situations
- B) The input data is corrupted
- C) The computer needs to be restarted
- D) The programming language is incompatible

42. What is a potential reason for a program producing incorrect output?
- A) The algorithm was not properly converted into a program
- B) The computer is outdated
- C) The user did not enter the data quickly enough

- D) The program was written in an unsupported language

43. What are problems with a program that prevent it from running correctly known as?
- A) Glitches
- B) Bugs
- C) Errors
- D) Faults

44. What should be done if the program produces an incorrect solution due to a bug?
- A) Restart the computer
- B) Debug the program and correct the algorithm if necessary
- C) Change the programming language
- D) Increase the computer's memory

45. After ensuring a program produces the correct result, what should be the next step?
- A) Delete the program
- B) Reconsider the original problem to ensure the solution is formatted correctly
- C) Publish the results immediately
- D) Stop the program from running again

46. Why might a solution need to be reformatted after a program produces correct results?
- A) To make the program run faster
- B) To ensure the output is presented in a way that solves the problem as intended
- C) To reduce the file size
- D) To translate the output into different languages

47. What might be needed if the program's output is a long list of numbers, but the goal was to identify a pattern?
- A) Print out the numbers on paper
- B) Display the information in a chart or graph
- C) Convert the list into a text file
- D) Re-run the program with the same input

48. When evaluating a solution, what might be discovered that could lead to additional steps being required?
- A) The program is too simple
- B) The code needs to be longer
- C) Additional data is needed to fully solve the problem
- D) The program needs to be translated into another language

49. What is the primary responsibility of the problem solver after the program produces results?
- A) To interpret the results in a meaningful way and determine if the problem is solved
- B) To upgrade the computer's hardware
- C) To publish the code online

- D) To immediately delete the source code

50. What action might be necessary if the program's results do not fully solve the problem?
- A) Re-do some of the problem-solving steps, possibly starting from step 1
- B) Ignore the results and try a different problem
- C) Increase the input data size
- D) Rewrite the program in a different programming language

51. Why is it important to interpret the results after running a program?
- A) To check for syntax errors
- B) To ensure the solution aligns with the original problem and objectives
- C) To see how fast the program runs
- D) To find hidden features in the code

52. If the results of a program indicate that some instructions were performed out of sequence, what should be checked?
- A) The computer's operating system
- B) The algorithm and the sequence of steps in the program
- C) The size of the input file
- D) The color of the output text

53. What does the term "compiles" refer to in the context of programming?
- A) The process of converting source code into machine code
- B) The process of collecting input data
- C) The process of saving a file
- D) The process of writing pseudo-code

54. What could happen if a bug in a program is not identified and fixed?
- A) The program may continue to produce incorrect results
- B) The program will run faster
- C) The computer may crash
- D) The program will become unusable

Answers
1. C
2. B
3. C
4. B
5. B
6. C
7. B
8. B
9. B
10. C

11. C
12. C
13. B
14. C
15. B
16. B
17. B
18. C
19. B
20. B
21. C
22. B
23. B
24. C
25. B
26. B
27. B
28. B
29. B
30. A
31. B
32. B
33. A
34. B
35. A
36. B
37. B
38. A
39. A
40. B
41. A
42. A
43. B
44. B
45. B
46. B
47. B
48. C
49. A
50. A
51. B
52. B
53. A

Module 2
M1_u1

1. What is an algorithm design technique?
   - A) A way to write code
   - B) A tool for debugging
   - C) A general approach to solving problems algorithmically
   - D) A method to organize data

2. Which of the following is an example of an algorithm design technique?
   - A) Sorting
   - B) Data storage
   - C) Brute-force
   - D) File handling

3. Why are algorithm design techniques important?
   - A) They help classify problems based on difficulty
   - B) They provide guidance in designing algorithms for new problems
   - C) They are only used for solving mathematical problems
   - D) They make programming languages more powerful

4. Which algorithm design technique involves trying all possible solutions?
   - A) Dynamic programming
   - B) Brute-force
   - C) Backtracking
   - D) Branch and bound

5. Why might brute-force techniques be inefficient for large problems?
   - A) They require complex mathematical calculations
   - B) They involve minimal programming effort
   - C) They exhaustively check all possible solutions
   - D) They are dependent on specific problem heuristics

6. What is the key characteristic of brute-force algorithms that makes them reliable for finding a solution?
   - A) They use advanced mathematical strategies
   - B) They check only the most likely solutions
   - C) They guarantee finding a solution if one exists
   - D) They optimize the solution process

7. Which of the following is an example where brute-force technique is used?
   - A) Data compression
   - B) Password cracking

- C) Machine learning
- D) Image processing

8. In which problem does brute-force check every possible route to find the shortest path?
   - A) Knapsack problem
   - B) Traveling Salesman Problem
   - C) Sorting problem
   - D) Graph coloring

9. What is the main advantage of brute-force algorithms?
   - A) They are very efficient for all problem sizes
   - B) They are widely applicable to different types of problems
   - C) They use minimal computing resources
   - D) They require complex implementations

10. What makes brute-force algorithms simple to implement?
   - A) They require advanced knowledge of the problem
   - B) They involve systematic checks of all possibilities
   - C) They minimize the number of operations
   - D) They use shortcuts to find solutions faster

11. How does the brute-force technique handle string matching problems?
   - A) By comparing only the first and last characters
   - B) By checking every possible starting position in the string
   - C) By using advanced data structures
   - D) By ignoring case sensitivity

12. Which sorting algorithm is based on the brute-force concept?
   - A) Quick Sort
   - B) Selection Sort
   - C) Merge Sort
   - D) Bubble Sort

13. What is the key difference between Selection Sort and Insertion Sort?
   - A) Selection Sort is faster than Insertion Sort
   - B) Insertion Sort places an unsorted element in its suitable place in each iteration
   - C) Selection Sort requires more memory
   - D) Insertion Sort is a recursive algorithm

14. Why might brute-force algorithms serve as the starting point for improved algorithms?
   - A) They are complex and hard to understand
   - B) They require minimal initial effort to develop
   - C) They are only applicable to simple problems
   - D) They are guaranteed to be the most efficient

15. What is a major drawback of brute-force algorithms for large input sizes?
- A) They are difficult to implement
- B) They become inefficient due to the large number of possibilities
- C) They often miss the correct solution
- D) They require specialized hardware

16. Which of the following is NOT an advantage of the brute-force technique?
- A) Simplicity in implementation
- B) Guaranteed solution if one exists
- C) Applicability to a wide range of problems
- D) High efficiency for large problems

17. How does brute-force handle problems with exponential growth in possibilities?
- A) It skips certain possibilities to save time
- B) It checks every possibility, which makes it inefficient
- C) It uses a divide and conquer approach
- D) It relies on random sampling to find the solution

18. In what scenario might brute-force algorithms be most appropriate?
- A) When a quick, simple solution is needed for a small problem
- B) When optimizing a solution for large, complex problems
- C) When using machine learning techniques
- D) When dealing with real-time processing

19. Why are brute-force algorithms not always the best choice for solving problems?
- A) They are too complex for simple problems
- B) They often require specialized algorithms
- C) They can be inefficient for large input sizes
- D) They are not applicable to all types of problems

20. Which of the following scenarios is likely to be inefficient when using brute-force?
- A) Small-scale password cracking
- B) Sorting a list of 10 elements
- C) Traveling Salesman Problem with 20 cities
- D) String matching in a short text

Answers:

1. C
2. C
3. B
4. B
5. C

6. C
7. B
8. B
9. B
10. B
11. B
12. B&D
13. B
14. B
15. B
16. D
17. B
18. A
19. C
20. C

M2_u2

1. What is the basic principle of the divide and conquer technique?
   - A) Combining small problems into one big problem
   - B) Dividing a big problem into sub-problems and solving them
   - C) Iteratively solving a problem in a loop
   - D) Solving problems using brute force

2. In the divide and conquer technique, what is the first step?
   - A) Combine solutions of sub-problems
   - B) Divide the problem into sub-problems
   - C) Conquer the sub-problems by solving them
   - D) Identify the base case

3. Which of the following sorting algorithms is based on the divide and conquer technique?
   - A) Bubble Sort
   - B) Selection Sort
   - C) Merge Sort
   - D) Insertion Sort

4. What is the purpose of the "conquer" step in divide and conquer?
   - A) To solve the original problem
   - B) To solve the sub-problems
   - C) To combine the solutions of sub-problems
   - D) To divide the sub-problems further

5. Which of the following is not a characteristic of the divide and conquer technique?
   - A) Recursion
   - B) Efficiency

- C) Parallelism
- D) Exhaustive search

6. What does the divide and conquer technique generally improve in algorithms?
- A) Space complexity
- B) Time complexity
- C) Readability
- D) Flexibility

7. Which algorithm is used for efficient searching in a sorted array using divide and conquer?
- A) Linear Search
- B) Binary Search
- C) Depth-First Search
- D) Breadth-First Search

8. The Karatsuba algorithm, which is used for multiplying large integers, is an example of:
- A) Brute force
- B) Divide and conquer
- C) Dynamic programming
- D) Backtracking

9. What is the main advantage of parallelism in the divide and conquer technique?
- A) It reduces the number of sub-problems
- B) It simplifies the problem
- C) It allows sub-problems to be solved simultaneously
- D) It reduces memory usage

10. Which of the following is a disadvantage of the divide and conquer technique?
- A) Simplicity
- B) Improved Efficiency
- C) Overhead from recursive calls
- D) Easy parallelization

11. Strassen's algorithm, which is used for matrix multiplication, reduces the number of:
- A) Additions
- B) Subtractions
- C) Multiplicative operations
- D) Divisions

12. Which step of divide and conquer is responsible for generating the solution to the original problem?
- A) Divide
- B) Conquer
- C) Combine

- D) Optimize

13. The divide and conquer technique is particularly well-suited for problems that:
  - A) Can be solved iteratively
  - B) Have overlapping sub-problems
  - C) Can be divided into independent sub-problems
  - D) Require a greedy approach

14. Which sorting algorithm divides the list into two halves, recursively sorts each half, and then merges the sorted halves?
  - A) Quick Sort
  - B) Merge Sort
  - C) Heap Sort
  - D) Bubble Sort

15. The divide and conquer approach is often more efficient than iterative methods because:
  - A) It always reduces space complexity
  - B) It has a lower overhead
  - C) It reduces the size of the problem at each step
  - D) It doesn't require recursion

Answers

1. B
2. B
3. C
4. B
5. D
6. B
7. B
8. B
9. C
10. C
11. C
12. C
13. C
14. B
15. C

Module 3

M3_u1

1. What does a flowchart represent in algorithm design?
  - A) Mathematical equations
  - B) Graphical representation of an algorithm

- C) Pseudo-code implementation
- D) Written computer programs

2. Which of the following symbols is used to represent a process in a flowchart?
- A) Oval
- B) Diamond
- C) Rectangle
- D) Parallelogram

3. In flowcharting, what does a diamond shape represent?
- A) Process
- B) Decision
- C) Input/Output
- D) Terminal

4. Which symbol in a flowchart is used to indicate the start and stop of the chart?
- A) Circle
- B) Oval
- C) Rectangle
- D) Parallelogram

5. What is the purpose of the parallelogram symbol in a flowchart?
- A) To represent a process
- B) To indicate input/output
- C) To show the flow of control
- D) To mark the end of the program

6. Which flowchart symbol is used to connect flowcharts that span across multiple pages?
- A) Diamond
- B) Rectangle
- C) Circle
- D) Arrow

7. What is the flow arrow or line used for in a flowchart?
- A) To start and stop the process
- B) To represent input and output operations
- C) To indicate the direction of the process flow
- D) To make decisions

8. Which of the following is NOT a rule recommended by the American National Standards Institute (ANSI) for flowcharting?
- A) Flowcharts must have only one starting and ending point
- B) Flowcharts can have multiple decision points with no connectors
- C) Flowcharts should follow a top-to-bottom or left-to-right approach

- D) Every aspect of the flowchart should use standard symbols

9. What does pseudo-code typically represent?
   - A) Fully executable code
   - B) A combination of English and programming syntax
   - C) Graphical representation of code
   - D) Mathematical formulas

10. Which of the following is true about pseudo-code?
   - A) It follows the syntax of a specific programming language
   - B) It can be compiled and executed directly
   - C) It provides a structured way to detail logic without specific programming syntax
   - D) It uses only graphical symbols to represent instructions

11. Which of the following programming languages can be used to write computer programs that implement algorithms?
   - A) Java
   - B) C++
   - C) PHP
   - D) All of the above

12. In a flowchart, which symbol should have two flow lines connecting to the previous and next symbols?
   - A) Oval
   - B) Rectangle
   - C) Parallelogram
   - D) Diamond

13. In the context of flowcharts, what is the purpose of the connector symbol (circle)?
   - A) To start the flowchart
   - B) To represent a decision point
   - C) To indicate the end of the flowchart
   - D) To connect different parts of the flowchart

14. Why are flowcharts and pseudo-code important in algorithm development?
   - A) They make algorithms easier to understand and implement
   - B) They are only used in advanced algorithm development
   - C) They are necessary for all types of programming languages
   - D) They replace the need for writing actual code

15. What should be the first step in developing a flowchart or pseudo-code for an algorithm?
   - A) Start coding the solution in a programming language
   - B) Identify the problem and the main steps to solve it
   - C) Create a complex design without considering the problem

- D) Execute the solution to see the results

Answers:

1. B
2. C
3. B
4. B
5. B
6. C
7. C
8. B
9. B
10. C
11. D
12. D
13. D
14. A
15. B

M3_u2

1. What is a decision table used for in computer science?
   - A) Compiling code
   - B) Graphically representing an algorithm for decision-making
   - C) Writing pseudocode
   - D) Implementing recursion

2. Which of the following is NOT a component of a decision table?
   - A) Condition/Attribute Component
   - B) Action/Conclusion Component
   - C) Data/Output Component
   - D) Condition/Rule Applied Component

3. What does the Condition/Attribute Component in a decision table represent?
   - A) The actions to be taken
   - B) The conditions or attributes based on which a decision is made
   - C) The possible outcomes
   - D) The rules applied to a decision

4. In a decision table, what does the Action/Conclusion Component specify?
   - A) The conditions under which decisions are made
   - B) The rules that need to be applied

- C) The actions to be taken based on specific conditions
- D) The final decision taken

5. Which component in a decision table contains the conclusion or decision selected from possible outcomes?
- A) Condition/Attribute Component
- B) Condition/Rule Applied Component
- C) Action/Conclusion Component
- D) Action/Conclusion/Decision Taken Component

6. What does a decision tree primarily represent?
- A) A set of conditions for decision-making
- B) A list of potential outcomes
- C) The sequence of actions in a flowchart
- D) The structure of a recursive algorithm

7. Which symbol is used to represent a condition or attribute node in a decision tree?
- A) Rectangle
- B) Arrow
- C) Oval
- D) Diamond

8. In a decision tree, what does a rectangle symbol represent?
- A) A condition or attribute node
- B) A decision, conclusion, or leaf node
- C) A start node
- D) A process block

9. What is the starting point of a decision tree called?
- A) Leaf node
- B) Root node
- C) Conclusion node
- D) Condition node

10. How does a decision tree process decisions?
- A) By looping back to previous nodes
- B) By starting from the root node and traversing to the leaf node
- C) By using a recursive function
- D) By combining conditions and actions in a table

11. In a decision tree, what does the arrow symbol indicate?
- A) A process block
- B) The flow of control or direction of the branch
- C) The final decision

- D) The end of the algorithm

12. Which of the following best describes the relationship between decision tables and decision trees?
- A) Both are used exclusively for mathematical operations
- B) Both are graphical representations for modeling decision processes
- C) Decision tables are a type of decision tree
- D) Decision trees are used to replace decision tables

13. Which of the following is an advantage of using decision tables?
- A) They provide a visual representation of decision-making
- B) They eliminate the need for coding
- C) They are easier to implement than decision trees
- D) They automatically generate solutions

14. What is the final outcome of a decision tree referred to as?
- A) Root node
- B) Leaf node
- C) Condition node
- D) Branch node

15. Which tool is particularly useful for modeling and analyzing decision processes in complex systems?
- A) Flowchart
- B) Decision table
- C) Pseudocode
- D) Recursive algorithm

Answers:

1. B
2. C
3. B
4. C
5. D
6. A
7. C
8. B
9. B
10. B
11. B
12. B
13. A
14. B

15. B

Module 4

M4_u1

1. What is the primary goal of sorting in computing?
   - A) Rearranging items in a list for better visualization
   - B) Rearranging the items of a given list in ascending or descending order
   - C) Counting the number of items in a list
   - D) Finding duplicates in a list

2. Which of the following is NOT an example of a sorting algorithm?
   - A) Bubble sort
   - B) Merge sort
   - C) Binary search
   - D) Insertion sort

3. Why is sorting important in computing?
   - A) It helps in faster data processing
   - B) It makes data look organized
   - C) It enables faster solution to problems like searching
   - D) It helps in data compression

4. What is searching in computing?
   - A) Rearranging items in a list
   - B) Finding a given value in a set
   - C) Deleting items from a list
   - D) Merging two lists

5. Which of the following is an example of a searching algorithm?
   - A) Quick sort
   - B) Sequential search
   - C) Merge sort
   - D) Bubble sort

6. In which scenario is binary search particularly effective?
   - A) On unsorted data
   - B) On small data sets
   - C) On sorted data
   - D) On data with duplicates

7. What is the key difference between binary search and linear search?
   - A) Binary search is faster but can only be used on sorted data
   - B) Linear search is faster but can only be used on sorted data

- C) Binary search can find multiple items at once
- D) Linear search always starts from the middle of the list

8. Which algorithm design technique does Bubble sort fall under?
- A) Divide and conquer
- B) Greedy algorithm
- C) Dynamic programming
- D) Brute force

9. Which of the following sorting algorithms is based on the divide and conquer technique?
- A) Selection sort
- B) Bubble sort
- C) Quick sort
- D) Insertion sort

10. What is an example of a problem that can be solved using the brute force design technique?
- A) Quick sort
- B) Sequential search
- C) Binary search
- D) Strassen's matrix multiplication

11. Which sorting algorithm is most suitable for sorting a small list of numbers with minimal computational resources?
- A) Bubble sort
- B) Merge sort
- C) Quick sort
- D) Radix sort

12. In which computing problem area would you most likely use Strassen's matrix multiplication?
- A) Graph traversal
- B) String processing
- C) Numerical computation
- D) Sorting

13. Which of the following is an application of graph algorithms?
- A) Bubble sort
- B) Binary search
- C) Binary tree traversal
- D) Text pattern matching

14. Which algorithm design technique is most commonly used in text pattern matching?
- A) Divide and conquer
- B) Greedy algorithm
- C) Brute force

- D) Dynamic programming

15. What is the primary advantage of using binary search over sequential search?
  - A) Binary search can be used on unsorted data
  - B) Binary search is more memory-efficient
  - C) Binary search is faster on sorted data
  - D) Binary search can search for multiple items simultaneously

Answers:

1. B
2. C
3. C
4. B
5. B
6. C
7. A
8. D
9. C
10. B
11. A
12. C
13. C
14. C
15. C

M4_u2

1. Which of the following operations is primarily used to determine the order of elements in a sorting algorithm?
  - A) Copy Operation
  - B) Comparison Operation
  - C) Swap Operation
  - D) Merge Operation

2. In the Simple Sort algorithm, what happens during the first pass?
  - A) The largest element is moved to the beginning of the list.
  - B) The smallest element is moved to the sorted portion of the list.
  - C) All elements are swapped with each other.
  - D) The list is divided into two equal parts.

3. Which sorting algorithm divides the array into smaller subarrays before sorting?
  - A) Selection Sort
  - B) Insertion Sort
  - C) Simple Sort

- D) Merge Sort

4. During which operation is a temporary cell created in the computer to facilitate the sorting process?
- A) Comparison Operation
- B) Copy Operation
- C) Swap Operation
- D) Merge Operation

5. Which sorting algorithm repeatedly selects the smallest element from the unsorted portion of the list and moves it to the sorted portion?
- A) Insertion Sort
- B) Merge Sort
- C) Selection Sort
- D) Quick Sort

6. In the Insertion Sort algorithm, where is the first unsorted element placed after being compared?
- A) At the end of the sorted portion
- B) At the beginning of the unsorted portion
- C) In its correct sorted position
- D) In the middle of the unsorted portion

7. Which sorting algorithm is characterized by the process of repeatedly dividing the array into halves until individual elements are reached, and then merging them back in sorted order?
- A) Selection Sort
- B) Bubble Sort
- C) Insertion Sort
- D) Merge Sort

8. What is the main difference between the Simple Sort and Selection Sort algorithms?
- A) Simple Sort uses comparison, while Selection Sort uses swapping.
- B) Simple Sort repeatedly moves the smallest element to the sorted portion, while Selection Sort selects the smallest element and swaps it with the first unsorted element.
- C) Simple Sort is a type of Merge Sort, while Selection Sort is a type of Insertion Sort.
- D) Simple Sort uses multiple arrays, while Selection Sort uses a single array.

9. In which of the following sorting algorithms does each element get inserted into its correct position within the sorted portion of the array one at a time?
- A) Quick Sort
- B) Selection Sort
- C) Bubble Sort
- D) Insertion Sort

10. Which sorting algorithm is most efficient for nearly sorted lists?
    - A) Bubble Sort
    - B) Quick Sort
    - C) Insertion Sort
    - D) Selection Sort

Answers

1. B
2. B
3. D
4. C
5. C
6. C
7. D
8. B
9. D
10. C