

Data Structures and Algorithms

Multiple Choice Questions (100 Questions)

Instructions:

- Each question has 4 options (A, B, C, D)
 - Choose the best answer for each question
 - Time: 120 minutes
-

Unit 1: Introduction to Data Structures & Algorithms (8 Questions)

1. **What is an Abstract Data Type (ADT)?** A. A data type that can only store integers B. A mathematical model of data objects with associated operations C. A data structure implemented in assembly language D. A type that cannot be modified once created
 2. **Which of the following is NOT a characteristic of an algorithm?** A. Finiteness B. Ambiguity C. Effectiveness D. Input/Output
 3. **What does the Big-O notation primarily describe?** A. Exact running time of an algorithm B. Worst-case time complexity C. Best-case time complexity D. Memory usage in bytes
 4. **Which asymptotic notation represents an upper bound that is not tight?** A. Θ (Theta) B. Ω (Omega) C. O (Big-O) D. ω (Small-omega)
 5. **What is the time complexity of accessing an element in an array?** A. $O(n)$ B. $O(\log n)$ C. $O(1)$ D. $O(n^2)$
 6. **Which function grows the slowest?** A. 2^n B. $n!$ C. n^3 D. $n \log n$
 7. **In C, which function is used for dynamic memory allocation?** A. `malloc()` B. `allocate()` C. `new()` D. `create()`
 8. **What does the term "data abstraction" refer to?** A. Hiding implementation details and showing only essential features B. Converting data to abstract art C. Removing all data from memory D. Making data structures more complex
-

Unit 2: Stack (8 Questions)

9. **Which principle does a Stack follow?** A. First In First Out (FIFO) B. Last In First Out (LIFO) C. Random Access D. Priority Based
 10. **Which operation adds an element to the top of the stack?** A. Pop B. Push C. Peek D. Pull
 11. **What is the time complexity of push and pop operations in a stack implemented using an array?** A. $O(n)$ B. $O(\log n)$ C. $O(1)$ D. $O(n^2)$
 12. **Which application typically uses a stack data structure?** A. Printer queue B. Function call management C. CPU scheduling D. Hash table
 - *13. **The postfix expression " $5\ 3\ +\ 8$ " evaluates to:** A. 64 B. 56 C. 19 D. 40
 14. **Which of the following is the postfix form of " $A+B*C$ "?** A. `ABC*+` B. `A+BC*` C. `AB+C*` D. `ABC+*`
 15. **What is the main disadvantage of array implementation of stack?** A. Poor time complexity B. Fixed size limitation C. Difficult implementation D. Cannot store different data types
 16. **Which stack operation returns the top element without removing it?** A. Pop B. Push C. Peek/Top D. `isEmpty`
-

Unit 3: Queue (8 Questions)

17. **Which principle does a Queue follow?** A. First In First Out (FIFO) B. Last In First Out (LIFO) C. Random Access D. Priority Based
 18. **In a circular queue, how is the front calculated after dequeuing an element?** A. `front = front + 1` B. `front = (front + 1) % capacity` C. `front = front - 1` D. `front = 0`
 19. **Which type of queue processes elements based on priority?** A. Linear Queue B. Circular Queue C. Priority Queue D. Double-ended Queue
 20. **What problem occurs in a simple array implementation of linear queue?** A. Stack overflow B. Memory leak C. Queue underflow D. Cannot utilize empty spaces after dequeues
 21. **Which operation removes an element from the queue?** A. Enqueue B. Dequeue C. Peek D. Push
 22. **In a circular queue, the condition `(rear+1)%capacity == front` indicates:** A. Queue is empty B. Queue has one element C. Queue is full D. Queue is half full
 23. **Which real-world scenario best represents a queue data structure?** A. Pile of plates B. Ticket counter line C. Family tree D. Dictionary lookup
 24. **What is the time complexity of enqueue and dequeue operations in a queue implemented using linked list?** A. $O(n)$ B. $O(\log n)$ C. $O(1)$ D. $O(n^2)$
-

Unit 4: Recursion (6 Questions)

25. **What is the base case in recursion?** A. The most complex case B. The case that stops the recursion C. The first function call D. The recursive call
26. **Which problem is commonly solved using recursion?** A. Linear search B. Fibonacci sequence C. Bubble sort D. Array traversal
27. **What is tail recursion?** A. Recursive call is the last operation in the function B. Multiple recursive calls C. Recursion without parameters D. Recursive call with loop

28. For the Tower of Hanoi problem with n disks, how many moves are required? A. n B. n^2 C. $2^n - 1$ D. $n!$
29. What is a potential disadvantage of recursion compared to iteration? A. Less readable code B. Higher time complexity C. Stack overflow for deep recursion D. More difficult to debug
30. The recursive Fibonacci algorithm has time complexity of: A. $O(n)$ B. $O(\log n)$ C. $O(2^n)$ D. $O(n^2)$
-

Unit 5: Lists (16 Questions)

31. What is a linked list node composed of? A. Only data B. Data and pointer to next node C. Data and two pointers D. Pointer only
32. In a singly linked list, how is the last node identified? A. Its next pointer points to NULL B. Its data is 0 C. Its next pointer points to the first node D. It has no data
33. What is the time complexity to insert a node at the beginning of a singly linked list? A. $O(n)$ B. $O(\log n)$ C. $O(1)$ D. $O(n^2)$
34. Which linked list allows traversal in both directions? A. Singly Linked List B. Doubly Linked List C. Circular Linked List D. Linear Linked List
35. In a circular linked list, the last node points to: A. NULL B. First node C. Previous node D. Itself
36. What is the main advantage of linked list over array? A. Constant time access to elements B. Dynamic size C. Less memory usage D. Cache friendly
37. To delete a node from a singly linked list, what information is needed? A. Only the node to delete B. The node and its next node C. The previous node D. All nodes in the list
38. What is the time complexity to search an element in a singly linked list? A. $O(1)$ B. $O(\log n)$ C. $O(n)$ D. $O(n^2)$
39. How can a stack be implemented using linked list? A. Insert/delete at beginning B. Insert at end, delete from beginning C. Insert at beginning, delete from end D. Insert/delete at end
40. What is a sentinel node in linked list? A. A node with maximum value B. A dummy node that simplifies operations C. The first data node D. A node that points to itself
41. Which operation is more efficient in array than linked list? A. Insertion at beginning B. Random access by index C. Deletion at end D. Sequential traversal
42. In a doubly linked list, each node contains: A. One data and one pointer B. One data and two pointers C. Two data and one pointer D. Two data and two pointers
43. What is the time complexity to insert at the end of a singly linked list (without tail pointer)? A. $O(1)$ B. $O(\log n)$ C. $O(n)$ D. $O(n^2)$
44. Which list implementation is better when frequent insertions/deletions at beginning are needed? A. Array B. Linked List C. Both are equally good D. Neither
45. Memory for a new node in linked list is allocated: A. At compile time B. At runtime C. From stack D. From register
46. What is a self-referential structure? A. Structure containing pointer to itself type B. Structure with no pointers C. Structure with array of itself D. Structure with only primitive types
-

Unit 6: Sorting (16 Questions)

47. Which sorting algorithm has worst-case time complexity $O(n^2)$ and best-case $O(n)$? A. Selection Sort B. Bubble Sort C. Insertion Sort D. Merge Sort
48. Which is NOT a comparison-based sorting algorithm? A. Quick Sort B. Merge Sort C. Radix Sort D. Heap Sort
49. What is the time complexity of merge sort? A. $O(n)$ B. $O(n \log n)$ C. $O(n^2)$ D. $O(\log n)$
50. In bubble sort, after the first pass: A. Smallest element is at its correct position B. Largest element is at its correct position C. Array is completely sorted D. No element is at correct position
51. Which sorting algorithm uses the divide and conquer strategy? A. Bubble Sort B. Selection Sort C. Insertion Sort D. Quick Sort
52. What is the main disadvantage of selection sort? A. Not stable B. High time complexity even when array is sorted C. Requires extra memory D. Difficult to implement
53. Which sorting algorithm has the best worst-case time complexity? A. Bubble Sort B. Quick Sort C. Merge Sort D. Insertion Sort
54. Heap sort is based on which data structure? A. Queue B. Stack C. Binary Heap D. Linked List
55. In quick sort, the element chosen for partitioning is called: A. Pivot B. Partition C. Divisor D. Splitter
56. Which sorting algorithm is in-place and has $O(n \log n)$ average case? A. Merge Sort B. Heap Sort C. Bubble Sort D. Selection Sort
57. What is the time complexity of shell sort in best case? A. $O(n)$ B. $O(n \log n)$ C. $O(n^2)$ D. $O(n^{4/3})$
58. Which sorting algorithm is stable? A. Quick Sort B. Heap Sort C. Merge Sort D. Selection Sort
59. External sorting is used when: A. Data fits in RAM B. Data is too large to fit in RAM C. Sorting small arrays D. Real-time sorting needed
60. In insertion sort, elements are: A. Compared and swapped with all previous elements B. Inserted at correct position in sorted portion C. Selected and placed at beginning D. Divided into two halves
61. Which is an adaptive sorting algorithm? A. Selection Sort B. Bubble Sort C. Merge Sort D. Heap Sort
62. What is the space complexity of quick sort in worst case? A. $O(1)$ B. $O(\log n)$ C. $O(n)$ D. $O(n^2)$
-

Unit 7: Searching and Hashing (12 Questions)

63. Binary search requires the array to be: A. Unsorted B. Sorted C. Partially sorted D. Reverse sorted

64. What is the time complexity of binary search? A. $O(1)$ B. $O(\log n)$ C. $O(n)$ D. $O(n^2)$
65. Sequential search has time complexity of: A. $O(1)$ B. $O(\log n)$ C. $O(n)$ D. $O(n^2)$
66. Which is NOT a collision resolution technique in hashing? A. Chaining B. Linear Probing C. Quadratic Probing D. Binary Search
67. A good hash function should: A. Minimize collisions B. Maximize collisions C. Be complex to compute D. Always return same value
68. In separate chaining, collisions are resolved by: A. Storing in next available slot B. Creating linked list at hash index C. Rehashing D. Discarding new element
69. Load factor in hash table is defined as: A. Number of slots / Number of elements B. Number of elements / Number of slots C. Size of table in bytes D. Number of collisions
70. Which probing method can cause secondary clustering? A. Linear Probing B. Quadratic Probing C. Double Hashing D. Chaining
71. Double hashing uses: A. One hash function B. Two different hash functions C. Same hash function twice D. No hash function
72. What is the best-case time complexity for search in hash table with chaining? A. $O(1)$ B. $O(\log n)$ C. $O(n)$ D. $O(n^2)$
73. Which search algorithm is more efficient for sorted array? A. Linear Search B. Binary Search C. Both are equally efficient D. Neither
74. The mid index in binary search is calculated as: A. $(\text{low} + \text{high}) / 2$ B. $(\text{high} - \text{low}) / 2$ C. $\text{low} + (\text{high} - \text{low}) / 2$ D. $\text{high} - \text{low}$
-

Unit 8: Trees and Graphs (16 Questions)

75. A binary tree has at most how many children per node? A. 1 B. 2 C. 3 D. Any number
76. Which tree traversal visits root, left subtree, then right subtree? A. Inorder B. Preorder C. Postorder D. Level order
77. In a Binary Search Tree, all nodes in left subtree are: A. Greater than root B. Less than root C. Equal to root D. Unordered
78. The height of a tree with single node is: A. 0 B. 1 C. 2 D. -1
79. An AVL tree is: A. A binary tree B. A balanced binary search tree C. A complete binary tree D. A full binary tree
80. Which rotation is used when left-left imbalance occurs in AVL tree? A. Left rotation B. Right rotation C. Left-right rotation D. Right-left rotation
81. What is the minimum number of nodes in a binary tree of height h ? A. h B. $h+1$ C. $2h$ D. 2^h
82. Which graph representation uses adjacency matrix? A. 2D array B. Linked list C. Array of linked lists D. Stack
83. Dijkstra's algorithm finds: A. Minimum Spanning Tree B. Shortest path from source to all nodes C. Maximum flow D. Graph connectivity
84. Kruskal's algorithm for MST uses which data structure? A. Stack B. Queue C. Priority Queue D. Disjoint Set
85. Which graph traversal uses queue? A. Depth First Search B. Breadth First Search C. Both D. Neither
86. A complete binary tree with n nodes has height: A. $O(n)$ B. $O(\log n)$ C. $O(n^2)$ D. $O(1)$
87. Which is NOT a tree traversal method? A. Inorder B. Preorder C. Postorder D. Sideorder
88. Prim's algorithm for MST is: A. Greedy algorithm B. Divide and conquer C. Dynamic programming D. Backtracking
89. In graph, degree of a vertex is: A. Number of edges incident to it B. Number of vertices connected C. Height of vertex D. Distance from source
90. Which algorithm can detect cycles in graph? A. Dijkstra B. Kruskal C. DFS D. Prim
-

Laboratory Works/Crossover Concepts (10 Questions)

91. When implementing stack using linked list, push operation is done at: A. Beginning of list B. End of list C. Middle of list D. Any position
92. Which sorting algorithm is implemented recursively in laboratory? A. Bubble Sort B. Selection Sort C. Quick Sort D. Insertion Sort
93. In lab, binary search tree operations include: A. Insertion, deletion, traversal B. Only insertion C. Only traversal D. Only deletion
94. Dynamic memory allocation in C for linked list node uses: A. `malloc()` and `free()` B. `new()` and `delete()` C. `alloc()` and `dealloc()` D. `create()` and `destroy()`
95. Graph representation using adjacency list in lab typically uses: A. Array of linked lists B. 2D array C. Single linked list D. Stack
96. Which data structure would be best for implementing a priority queue in lab? A. Array B. Linked List C. Heap D. Stack
97. In lab, hashing implementation includes: A. Hash function and collision handling B. Only hash function C. Only collision handling D. Neither
98. When implementing queue using linked list, we maintain: A. Only front pointer B. Only rear pointer C. Both front and rear pointers D. No pointers
99. Recursive implementation in lab includes: A. Fibonacci, Factorial, Tower of Hanoi B. Only Fibonacci C. Only Factorial D. Only Tower of Hanoi
100. AVL tree rotation implementation in lab ensures: A. Tree remains balanced B. Tree becomes complete C. Tree height increases D. Tree becomes linear
-

END OF QUESTIONS