

Data Structure and Algorithm

UNIT 1: Introduction to Data Structures & Algorithms (1-12)

1. Which of the following is a collection of data elements with a logical relationship?
 - A. Data type
 - B. Data structure
 - C. Algorithm
 - D. Variable

2. An Abstract Data Type (ADT) defines:
 - A. Only data
 - B. Only operations
 - C. Data and operations together
 - D. Memory allocation

3. Which of the following is NOT a primitive data type?
 - A. int
 - B. float
 - C. char
 - D. stack

4. Dynamic memory allocation in C is done using:
 - A. malloc()
 - B. sizeof()
 - C. printf()
 - D. scanf()

5. Which function is used to deallocate dynamic memory in C?
 - A. delete()
 - B. remove()
 - C. free()
 - D. clear()

6. Which header file is required for dynamic memory allocation?
 - A. stdio.h
 - B. conio.h
 - C. stdlib.h
 - D. string.h

7. An algorithm must be:
- A. Finite
 - B. Ambiguous
 - C. Infinite
 - D. Hardware dependent
8. Which notation describes the upper bound of an algorithm?
- A. Ω (Omega)
 - B. Θ (Theta)
 - C. O (Big-O)
 - D. Σ (Sigma)
9. Best case complexity is represented by:
- A. Big-O
 - B. Big- Ω
 - C. Big- Θ
 - D. Log-O
10. Which of the following grows fastest?
- A. $O(n)$
 - B. $O(\log n)$
 - C. $O(n^2)$
 - D. $O(1)$
11. Time complexity depends on:
- A. Compiler
 - B. Input size
 - C. Output size
 - D. Operating system
12. Which complexity represents constant time?
- A. $O(n)$
 - B. $O(\log n)$
 - C. $O(1)$
 - D. $O(n \log n)$
13. Stack follows which principle?
- A. FIFO
 - B. LIFO
 - C. FILO
 - D. Random

14. Which operation inserts an element into a stack?

- A. Pop
- B. Push
- C. Peek
- D. Display

15. Which operation removes an element from a stack?

- A. Push
- B. Pop
- C. Insert
- D. Append

16. Stack overflow occurs when:

- A. Stack is empty
- B. Stack is full
- C. Memory leak happens
- D. Stack pointer is null

17. Stack underflow occurs when:

- A. Stack is full
- B. Stack is empty
- C. Stack pointer overflows
- D. Stack grows dynamically

18. Which application uses stack?

- A. CPU scheduling
- B. Expression evaluation
- C. BFS traversal
- D. Disk scheduling

19. Infix expression is:

- A. A+B
- B. AB+
- C. +AB
- D. A+B+

20. Postfix expression is:

- A. A+B
- B. AB+
- C. +AB
- D. (A+B)

21. Prefix expression is:

- A. A+B
- B. AB+
- C. +AB
- D. (A+B)

22. Stack is used in:

- A. Recursion
- B. Sorting
- C. Searching
- D. Hashing

23. Which element is accessed in stack?

- A. Bottom
- B. Middle
- C. Random
- D. Top

24. Which stack operation returns top element without removing it?

- A. Push
- B. Pop
- C. Peek
- D. Display

25. Queue follows which principle?

- A. LIFO
- B. FIFO
- C. FILO
- D. Random

26. In queue, insertion happens at:

- A. Front
- B. Middle
- C. Rear
- D. Any position

27. Deletion in queue happens at:

- A. Rear
- B. Front
- C. Middle
- D. End

28. Which queue overcomes wastage of space?

- A. Linear Queue
- B. Simple Queue
- C. Circular Queue
- D. Priority Queue

29. Which queue assigns priority to elements?

- A. Linear Queue
- B. Circular Queue
- C. Priority Queue
- D. Deque

30. Queue overflow occurs when:

- A. Queue is empty
- B. Queue is full
- C. Front = -1
- D. Rear = -1

31. Which application uses queue?

- A. Function call
- B. Recursion
- C. CPU scheduling
- D. Expression evaluation

32. Which operation inserts element in queue?

- A. Dequeue
- B. Enqueue
- C. Push
- D. Pop

33. Which operation removes element from queue?

- A. Enqueue
- B. Push
- C. Pop
- D. Dequeue

34. Deque allows insertion and deletion at:

- A. Front only
- B. Rear only
- C. Both ends
- D. Middle

35. Which queue is used in BFS?

- A. Stack
- B. Priority Queue
- C. Circular Queue
- D. Simple Queue

36. Queue can be implemented using:

- A. Array
- B. Linked list
- C. Both A and B
- D. None

37. A function calling itself is called:

- A. Iteration
- B. Recursion
- C. Looping
- D. Nesting

38. Base condition is used to:

- A. Increase recursion
- B. Stop recursion
- C. Optimize memory
- D. Increase speed

39. Tail recursion means:

- A. Recursive call at start
- B. Recursive call in middle
- C. Recursive call at end
- D. No recursive call

40. Which problem is best solved using recursion?

- A. Factorial
- B. Sorting
- C. Searching
- D. Hashing

41. Tower of Hanoi problem uses:

- A. Stack
- B. Queue
- C. Recursion
- D. Tree

42. Recursive calls are stored in:

- A. Heap
- B. Queue
- C. Stack
- D. Array

43. Recursive approach generally uses:

- A. Less memory
- B. More memory
- C. No memory
- D. Fixed memory

44. Which is faster generally?

- A. Recursion
- B. Iteration
- C. Both same
- D. Depends on compiler

45. Fibonacci sequence is an example of:

- A. Linear recursion
- B. Binary recursion
- C. Tail recursion
- D. Infinite recursion

46. GCD can be computed using:

- A. Stack
- B. Queue
- C. Recursion
- D. Hashing

47. Recursive function must have:

- A. Loop
- B. Return value
- C. Base case
- D. Global variable

48. Which is NOT an application of recursion?

- A. Tree traversal
- B. Graph traversal
- C. Binary search
- D. Bubble sort

49. A list is a collection of:

- A. Similar data
- B. Different data
- C. Random data
- D. Unrelated data

50. Which list implementation uses contiguous memory?

- A. Linked list
- B. Array list
- C. Circular list
- D. Doubly list

51. Linked list stores elements in:

- A. Contiguous memory
- B. Random memory
- C. Stack
- D. Queue

52. Each node in singly linked list contains:

- A. Data only
- B. Address only
- C. Data and next pointer
- D. Two pointers

53. Doubly linked list has:

- A. One pointer
- B. Two pointers
- C. Three pointers
- D. No pointer

54. Circular linked list last node points to:

- A. NULL
- B. First node
- C. Middle node
- D. Itself

55. Which insertion is fastest in linked list?

- A. Beginning
- B. Middle
- C. End
- D. Random

56. Which deletion is easiest in linked list?

- A. Beginning
- B. End
- C. Middle
- D. Random

57. Linked list overcomes:

- A. Stack overflow
- B. Memory wastage
- C. Queue overflow
- D. Recursion

58. Stack can be implemented using:

- A. Array
- B. Linked list
- C. Both
- D. None

59. Queue implemented using linked list avoids:

- A. Overflow
- B. Underflow
- C. Priority
- D. Recursion

60. Head pointer of linked list points to:

- A. Last node
- B. First node
- C. NULL
- D. Random

61. Which operation is costly in linked list?

- A. Insertion
- B. Deletion
- C. Traversal
- D. Memory allocation

62. Linked list does not allow:

- A. Dynamic size
- B. Efficient insertion
- C. Random access
- D. Efficient deletion

63. Which list uses both previous and next pointers?

- A. Singly
- B. Circular
- C. Doubly
- D. Simple

64. Which is NOT a type of linked list?

- A. Singly
- B. Doubly
- C. Circular
- D. Linear array

65. Sorting means arranging data in:

- A. Random order
- B. Ascending/Descending
- C. Reverse only
- D. Hash order

66. Bubble sort repeatedly:

- A. Selects minimum
- B. Swaps adjacent elements
- C. Divides array
- D. Builds heap

67. Selection sort selects:

- A. Largest element
- B. Smallest element
- C. Random element
- D. Middle element

68. Insertion sort works like:

- A. Playing cards
- B. Queue
- C. Stack
- D. Tree

69. Which sort is stable?

- A. Selection sort
- B. Quick sort
- C. Bubble sort
- D. Heap sort

70. Merge sort follows:

- A. Greedy method
- B. Dynamic programming
- C. Divide and conquer
- D. Backtracking

71. Quick sort uses:

- A. Pivot
- B. Heap
- C. Queue
- D. Stack only

72. Worst case of quick sort is:

- A. $O(n)$
- B. $O(n \log n)$
- C. $O(n^2)$
- D. $O(\log n)$

73. Heap sort uses:

- A. Binary tree
- B. Binary heap
- C. Stack
- D. Queue

74. Which sorting is fastest on average?

- A. Bubble
- B. Selection
- C. Quick
- D. Insertion

75. External sorting is used when:

- A. Data fits in memory
- B. Data is small
- C. Data is large
- D. Data is sorted

76. Shell sort is an improvement of:

- A. Bubble
- B. Selection
- C. Insertion
- D. Merge

77. Merge sort requires:

- A. No extra space
- B. Constant space
- C. Extra memory
- D. Stack only

78. Which is comparison sort?

- A. Counting sort
- B. Radix sort
- C. Bubble sort
- D. Bucket sort

79. Searching means:

- A. Sorting data
- B. Finding element
- C. Deleting data
- D. Inserting data

80. Linear search time complexity:

- A. $O(1)$
- B. $O(\log n)$
- C. $O(n)$
- D. $O(n^2)$

81. Binary search requires:

- A. Sorted array
- B. Unsorted array
- C. Linked list
- D. Stack

82. Best case of binary search:

- A. $O(n)$
- B. $O(n \log n)$
- C. $O(1)$
- D. $O(\log n)$

83. Hashing is used to:

- A. Sort data
- B. Search faster
- C. Traverse graph
- D. Compress data

84. Hash function maps keys to:

- A. Nodes
- B. Values
- C. Hash table index
- D. Queue

85. Collision occurs when:

- A. Same key inserted
- B. Same index generated
- C. Table is full
- D. Data lost

86. Which is collision resolution technique?

- A. Sorting
- B. Linear probing
- C. Traversal
- D. Recursion

87. Load factor is:

- A. Size of table
- B. Number of keys
- C. Keys/Table size
- D. Collisions

88. Which hashing method uses linked list?

- A. Linear probing
- B. Quadratic probing
- C. Chaining
- D. Rehashing

89. Tree is a:

- A. Linear structure
- B. Non-linear structure
- C. Circular structure
- D. Sequential structure

90. Root of tree has:

- A. Parent
- B. Child only
- C. No parent
- D. No child

91. Degree of a node is:

- A. Height
- B. Depth
- C. Number of children
- D. Level

92. Binary tree has maximum children:

- A. 1
- B. 2
- C. 3
- D. Unlimited

93. In order traversal of BST gives:

- A. Random order
- B. Descending order
- C. Ascending order
- D. Level order

94. AVL tree is:

- A. Unbalanced tree
- B. Balanced BST
- C. Heap tree
- D. Binary heap

95. Balance factor of AVL tree is:

- A. -2,0,2
- B. -1,0,1
- C. 0,1,2
- D. Any value

96. Graph consists of:

- A. Nodes only
- B. Edges only
- C. Vertices and edges
- D. Trees

97. BFS uses:

- A. Stack
- B. Queue
- C. Recursion
- D. Heap

98. DFS uses:

- A. Queue
- B. Stack
- C. Heap
- D. Array

99. Minimum spanning tree connects:

- A. All vertices with minimum cost
- B. Few vertices
- C. Only root
- D. Cycles

100. Dijkstra's algorithm is used for:

- A. Sorting
- B. Searching
- C. Shortest path
- D. Spanning tree

The END