

UNIT 1 - INTRODUCTION TO DATA STRUCTURES

Q1. A tree is a _____ data structure.

- a) Linear
- b) Non-linear
- c) Recursive
- d) None of these

Q2. Which of the following is a linear data structure?

- a) Stack
- b) Tree
- c) Graph
- d) None of these

Q3. The term ADT stands for _____.

- a) Abstract Data Type
- b) Advanced Data Tree
- c) Array Data Table
- d) Algorithm Design Type

Q4. Which data structure allows insertion and deletion from both ends?

- a) Queue
- b) Stack
- c) Deque
- d) Array

Q5. The time complexity of accessing an element in an array is _____.

- a) $O(1)$
- b) $O(n)$
- c) $O(\log n)$
- d) $O(n^2)$

Q6. What will be the output of the following code?

```
int arr[] = {5, 10, 15};  
cout << *(arr + 2);
```

- a) 5
- b) 10
- c) 15
- d) Compilation Error

Q7. In which of the following memory is data stored in continuous locations?

- a) Array
- b) Linked List
- c) Tree
- d) Graph

Q8. Which of the following is not a primitive data structure?

- a) Integer
- b) Float
- c) Stack
- d) Character

Q9. Which of the following operations is not possible in arrays?

- a) Traversal
- b) Insertion
- c) Deletion
- d) None of these

Q10. In an array of size n , the index of the last element is _____.

- a) n
- b) $n+1$
- c) $n-1$
- d) $n/2$

Q11. Which one is not a linear data structure?

- a) Array
- b) Stack
- c) Queue
- d) Binary Tree

Q12. Which of the following data structures uses LIFO order?

- a) Stack
- b) Queue
- c) Linked List
- d) Graph

Q13. Which of the following is an application of stack?

- a) Recursion
- b) Sorting
- c) Searching
- d) Hashing

Q14. A data structure that allows elements to be added or removed only at one end is _____.

- a) Queue
- b) Stack
- c) Array
- d) Graph

Q15. Fill in the blank: In linked list, elements are stored in _____ memory locations.

- a) Sequential
- b) Random
- c) Fixed
- d) Continuous

Q16. The process of arranging data in ascending or descending order is called _____.

- a) Searching
- b) Sorting
- c) Merging
- d) Traversing

Q17. The process of visiting each element in the data structure exactly once is called _____.

- a) Insertion
- b) Traversal
- c) Deletion
- d) Sorting

Q18. The data structure best suited for implementing recursion is _____.

- a) Array
- b) Stack
- c) Queue
- d) Graph

Q19. In linked list implementation, each node contains _____.

- a) Only data
- b) Only pointer
- c) Data and link
- d) None of these

Q20. Which of the following is a non-linear data structure?

- a) Array
- b) Stack
- c) Queue
- d) Graph

◆ ANSWER KEY - UNIT 1 (Q1-Q20)

1. b - A tree is a non-linear hierarchical structure.
2. a - Stack is a linear structure following LIFO order.
3. a - ADT means Abstract Data Type.
4. c - Deque allows insertion and deletion at both ends.
5. a - Accessing in an array takes constant time $O(1)$.
6. c - $*(arr + 2)$ accesses 3rd element, i.e., 15.
7. a - Arrays store elements in continuous memory.
8. c - Stack is a non-primitive structure.
9. d - All listed operations are possible.
10. c - Last index of array = $n - 1$.
11. d - Binary Tree is a non-linear structure.
12. a - Stack uses LIFO (Last In First Out).
13. a - Recursion is implemented using stack.
14. b - Stack allows insertion/deletion at one end (top).
15. b - Linked lists use random memory allocation.
16. b - Sorting arranges elements in order.
17. b - Traversal visits each element once.
18. b - Stack is used in recursion calls.
19. c - Node = data + pointer (link).
20. d - Graph is a non-linear data structure.

■ END OF SET 1 (QUESTIONS 1-20)

■ UNIT 1 - INTRODUCTION TO DATA STRUCTURES

(Questions 21-40)

Q21. Which of the following data structures follows FIFO order?

- a) Stack
- b) Queue
- c) Deque
- d) Linked List

Q22. The process of finding a specific element in a data structure is known as _____.

- a) Sorting
- b) Searching
- c) Traversal
- d) Merging

Q23. Which of the following is true about arrays?

- a) Fixed size
- b) Random access
- c) Elements stored contiguously
- d) All of the above

Q24. Fill in the blank: A _____ data structure is a collection of elements organized in a hierarchical manner.

- a) Stack
- b) Queue
- c) Tree
- d) Array

Q25. Which of the following is an example of a non-primitive data structure?

- a) int
- b) char
- c) array

d) float

Q26. Which data structure uses dynamic memory allocation?

- a) Array
- b) Linked List
- c) Stack (Array-based)
- d) Matrix

Q27. Which operation is used to remove an element from a stack?

- a) insert()
- b) pop()
- c) delete()
- d) remove()

Q28. A linked list is more efficient than an array in _____.

- a) Accessing elements
- b) Insertion and deletion
- c) Both a and b
- d) None of these

Q29. What will be the output of the following code?

```
int a[5] = {10, 20, 30, 40, 50};  
cout << *(a + 3);
```

- a) 10
- b) 20
- c) 30
- d) 40

Q30. The main disadvantage of a linked list is _____.

- a) Fixed size
- b) Extra memory for pointer
- c) Static structure
- d) Sequential allocation

Q31. Which of the following operations is not efficient in arrays?

- a) Traversing
- b) Insertion
- c) Access
- d) Display

Q32. Fill in the blank: A data structure is a way of _____ and _____ data.

- a) Storing, organizing
- b) Displaying, deleting
- c) Removing, sorting
- d) None of these

Q33. In stack implementation using array, overflow occurs when _____.

- a) Top = -1
- b) Top = size - 1
- c) Top = size
- d) None of these

Q34. In linked list implementation, underflow occurs when _____.

- a) Head = NULL
- b) Tail = NULL
- c) Data = 0
- d) None of these

Q35. Which of the following is not an application of queue?

- a) Printer spooler
- b) CPU scheduling
- c) Recursion
- d) Call center system

Q36. Which data structure allows elements to be added and removed alternately from both ends?

- a) Circular queue
- b) Stack
- c) Deque
- d) Linear queue

Q37. Which of the following data structures is best suited for evaluating arithmetic expressions?

- a) Queue
- b) Stack
- c) Tree
- d) Array

Q38. In a stack, if the sequence of operations is PUSH(1), PUSH(2), POP(), PUSH(3), POP(), the element removed in the last POP() is _____.

- a) 1
- b) 2
- c) 3
- d) None

Q39. Which of the following operations is performed first during recursion?

- a) Base condition check
- b) Recursive call
- c) Stack pop
- d) Stack push

Q40. In which case would you prefer a linked list over an array?

- a) When frequent insertion and deletion are required
- b) When random access is required
- c) When memory is limited and fixed
- d) When array indexing is needed

◆ ANSWER KEY - UNIT 1 (Q21-Q40)

- 21. b - Queue follows FIFO (First In First Out).
- 22. b - Searching means finding a specific element.
- 23. d - Arrays have fixed size, random access, and contiguous memory.
- 24. c - A tree stores data hierarchically.
- 25. c - Arrays are non-primitive, built from primitive types.
- 26. b - Linked lists use dynamic memory allocation.
- 27. b - pop() removes the top element in stack.
- 28. b - Linked list is efficient for insertion and deletion.
- 29. d - $*(a + 3)$ accesses the 4th element, which is 40.
- 30. b - Extra memory needed for storing pointers.
- 31. b - Insertion in arrays requires shifting → inefficient.
- 32. a - A data structure stores and organizes data efficiently.
- 33. b - Stack overflow when $top = size - 1$.
- 34. a - Linked list underflow when $head = NULL$ (empty).
- 35. c - Recursion uses stack, not queue.
- 36. c - Deque allows insertion and deletion from both ends.
- 37. b - Stack is used in arithmetic expression evaluation.
- 38. c - Last element popped is 3.
- 39. a - Base condition is checked first in recursion.
- 40. a - Linked list is preferred for frequent insertions/deletions.

■ END OF SET 2 (QUESTIONS 21-40)

■ UNIT 1 - INTRODUCTION TO DATA STRUCTURES

(Questions 41-60)

Q41. Which of the following data structures can be used to check for balanced parentheses in an expression?

- a) Queue
- b) Stack
- c) Array
- d) Tree

Q42. The number of null links in a linked list of n nodes is _____.

- a) n
- b) $n + 1$
- c) 1
- d) 0

Q43. The size of an array must be known at _____.

- a) Run time
- b) Compile time
- c) Execution time
- d) None of these

Q44. Which of the following traverses all elements exactly once?

- a) Traversal
- b) Sorting
- c) Searching
- d) Deletion

Q45. What is the output of the following code snippet?

```
int arr[3] = {2, 4, 6};  
cout << arr[0] + arr[2];
```

- a) 6
- b) 8
- c) 10
- d) 12

Q46. Which of the following statements is FALSE?

- a) Arrays have fixed size.
- b) Linked lists require contiguous memory.
- c) Stacks can be implemented using arrays.
- d) Queues can be implemented using linked lists.

Q47. Fill in the blank: In a stack, the last inserted element is the _____ to be removed.

- a) First
- b) Last
- c) Random
- d) Middle

Q48. Which of the following data structures is used in BFS (Breadth First Search)?

- a) Stack
- b) Queue
- c) Linked List
- d) Array

Q49. What is the disadvantage of using an array for insertion and deletion?

- a) Requires shifting of elements
- b) Easy to implement
- c) Fixed size
- d) Both a and c

Q50. In recursion, every function call is stored in _____.

- a) Stack

- b) Queue
- c) Array
- d) Tree

Q51. Fill in the blank: In a linked list, each node contains _____ and _____.

- a) Data, Address
- b) Value, Index
- c) Data, Data
- d) Key, Array

Q52. Which data structure allows random access to elements?

- a) Stack
- b) Array
- c) Linked List
- d) Queue

Q53. What will be the output of the code below?

```
int a[4] = {1, 2, 3, 4};  
cout << *(a + 1) + *(a + 3);
```

- a) 3
- b) 4
- c) 5
- d) 6

Q54. Which of the following operations cannot be performed on a queue efficiently using an array implementation?

- a) Insertion
- b) Deletion
- c) Display
- d) None

Q55. In an empty stack, the value of the top pointer is usually initialized to _____.

- a) 0
- b) 1
- c) -1
- d) NULL

Q56. The operation of adding an element to a queue is known as _____.

- a) Push
- b) Pop
- c) Enqueue
- d) Dequeue

Q57. Which of the following is not a valid operation on a stack?

- a) push()
- b) pop()
- c) enqueue()
- d) peek()

Q58. If a queue is implemented using a circular array, then after $\text{rear} = \text{size} - 1$, the next value of rear will be _____.

- a) 0
- b) 1
- c) size
- d) size + 1

Q59. What is the output of this code snippet?

```
int x = 5;  
int *p = &x;  
cout << *p + 5;
```

- a) 5
- b) 10

- c) 15
- d) Error

Q60. Fill in the blank: The logical or mathematical model of a data structure is called _____.

- a) Data Definition
- b) Abstract Data Type
- c) Data Model
- d) Structure Model

◆ ANSWER KEY - UNIT 1 (Q41-Q60)

- 41. b - Stack checks for balanced parentheses using LIFO order.
- 42. c - A singly linked list has one null link (last node).
- 43. b - Array size must be known at compile time.
- 44. a - Traversal visits all elements once.
- 45. c - $\text{arr}[0] + \text{arr}[2] = 2 + 6 = 8$ (Answer c) [Wait correction - correct answer is b=8, will fix below].
- 45. b - $\text{arr}[0] + \text{arr}[2] = 2 + 6 = 8$.
- 46. b - Linked lists do NOT require contiguous memory (FALSE statement).
- 47. a - In stack, last inserted is first to be removed (LIFO).
- 48. b - BFS uses Queue.
- 49. d - Both shifting and fixed size make arrays inefficient.
- 50. a - Recursive calls are stored in stack memory.
- 51. a - Node = data + address of next node.
- 52. b - Arrays allow random access via index.
- 53. d - $*(a + 1) = 2$, $*(a + 3) = 4 \rightarrow \text{sum} = 6$.
- 54. d - All operations can be done efficiently.
- 55. c - Empty stack top = -1.
- 56. c - Enqueue adds an element to queue.
- 57. c - enqueue() is for queue, not stack.
- 58. a - In circular queue, rear wraps to 0.
- 59. b - $*p = 5$, $*p + 5 = 10$.
- 60. b - Abstract Data Type defines logical model of data structure.

■ END OF SET 3 (QUESTIONS 41-60)

■ UNIT 1 - INTRODUCTION TO DATA STRUCTURES

(Questions 61-80)

Q61. The time complexity of linear search is _____.

- a) $O(1)$
- b) $O(\log n)$
- c) $O(n)$
- d) $O(n \log n)$

Q62. The best case time complexity of binary search is _____.

- a) $O(n)$
- b) $O(1)$
- c) $O(\log n)$
- d) $O(n^2)$

Q63. The average case complexity of binary search is _____.

- a) $O(n)$
- b) $O(1)$
- c) $O(\log n)$
- d) $O(n^2)$

Q64. The space complexity of a recursive algorithm includes space for _____.
a) Code
b) Variables
c) Function call stack
d) Input data

Q65. Which sorting algorithm has the best average case complexity?
a) Bubble Sort
b) Selection Sort
c) Merge Sort
d) Insertion Sort

Q66. The time complexity of Bubble Sort in the worst case is _____.
a) $O(n^2)$
b) $O(n \log n)$
c) $O(n)$
d) $O(1)$

Q67. The time complexity of Merge Sort is _____.
a) $O(n^2)$
b) $O(n \log n)$
c) $O(\log n)$
d) $O(n)$

Q68. The space complexity of Merge Sort is _____.
a) $O(1)$
b) $O(n)$
c) $O(n^2)$
d) $O(\log n)$

Q69. What is the worst-case time complexity of Quick Sort?
a) $O(n \log n)$
b) $O(n^2)$
c) $O(n)$
d) $O(\log n)$

Q70. Fill in the blank: Time complexity measures the _____ of an algorithm as input size increases.
a) Memory usage
b) Execution time
c) Code length
d) None

Q71. Which of the following algorithms is NOT comparison-based?
a) Merge Sort
b) Bubble Sort
c) Counting Sort
d) Insertion Sort

Q72. The Big O notation is used to represent _____.
a) Best case
b) Worst case
c) Average case
d) None of these

Q73. The time complexity of traversing a linked list of n elements is _____.
a) $O(1)$
b) $O(n)$
c) $O(n \log n)$
d) $O(n^2)$

Q74. Insertion at the beginning of a linked list takes _____ time.
a) $O(1)$
b) $O(n)$

- c) $O(\log n)$
- d) $O(n^2)$

Q75. The time complexity of inserting an element in an array of size n is _____.

- a) $O(1)$
- b) $O(n)$
- c) $O(\log n)$
- d) $O(n^2)$

Q76. Fill in the blank: For searching an element in an unsorted linked list, time complexity is _____.

- a) $O(1)$
- b) $O(\log n)$
- c) $O(n)$
- d) $O(n^2)$

Q77. What is the time complexity to delete a node from a singly linked list if pointer to the node is given?

- a) $O(1)$
- b) $O(n)$
- c) $O(\log n)$
- d) $O(n^2)$

Q78. The time complexity to search an element in a Binary Search Tree (BST) in the best case is _____.

- a) $O(1)$
- b) $O(\log n)$
- c) $O(n)$
- d) $O(n \log n)$

Q79. What is the time complexity for accessing an element in an array using its index?

- a) $O(1)$
- b) $O(n)$
- c) $O(\log n)$
- d) $O(n^2)$

Q80. Fill in the blank: Space complexity is the total _____ space required by an algorithm.

- a) Temporary and input
- b) Code
- c) Data
- d) Stack

 ♦ ANSWER KEY - UNIT 1 (Q61-Q80)

- 61. c – Linear search checks all elements $\rightarrow O(n)$.
- 62. b – Best case (found at mid) $\rightarrow O(1)$.
- 63. c – Average case for binary search $\rightarrow O(\log n)$.
- 64. c – Recursion adds function call stack frames.
- 65. c – Merge Sort has average $O(n \log n)$.
- 66. a – Bubble Sort worst case $O(n^2)$ (when array is reversed).
- 67. b – Merge Sort always runs in $O(n \log n)$.
- 68. b – Merge Sort needs extra memory $\rightarrow O(n)$.
- 69. b – Quick Sort worst case when pivot is poor $\rightarrow O(n^2)$.
- 70. b – Time complexity measures execution time growth.
- 71. c – Counting Sort is not comparison-based.
- 72. b – Big O shows asymptotic worst-case behavior.
- 73. b – Traversing linked list of n nodes takes $O(n)$.
- 74. a – Insert at beginning in $O(1)$.
- 75. b – Insertion in array (shift elements) $\rightarrow O(n)$.

76. c – Searching unsorted linked list $\rightarrow O(n)$.
77. a – If pointer given, delete node in $O(1)$.
78. b – Best case (root matches key) $\rightarrow O(\log n)$.
79. a – Array access via index is $O(1)$.
80. a – Space complexity = input + temporary space used.

☐ END OF SET 4 (QUESTIONS 61-80)

☐ UNIT 1 – INTRODUCTION TO DATA STRUCTURES

(Questions 81-100)

- Q81. The time complexity for traversing an array of size n is _____.
a) $O(1)$
b) $O(n)$
c) $O(n \log n)$
d) $O(n^2)$
- Q82. Which of the following sorting algorithms is the most efficient for large data sets?
a) Bubble Sort
b) Merge Sort
c) Selection Sort
d) Insertion Sort
- Q83. The recursive algorithm for factorial(n) has a time complexity of _____.
a) $O(n^2)$
b) $O(n)$
c) $O(\log n)$
d) $O(1)$
- Q84. Which of the following best describes space complexity?
a) Total time required by algorithm
b) Total memory required by algorithm
c) Number of variables used
d) Amount of data in input
- Q85. Fill in the blank: The Big Ω (Omega) notation represents the _____ bound of an algorithm.
a) Lower
b) Upper
c) Average
d) Middle
- Q86. What is the time complexity of the following loop?

```
for(int i=0; i<n; i++)  
    cout << i;
```


a) $O(1)$
b) $O(n)$
c) $O(n^2)$
d) $O(\log n)$
- Q87. What is the time complexity of nested loops below?

```
for(int i=0; i<n; i++)  
    for(int j=0; j<n; j++)  
        cout << i + j;
```


a) $O(n)$
b) $O(n \log n)$
c) $O(n^2)$
d) $O(n^3)$

Q88. What will be the output of this code?

```
int arr[] = {1, 2, 3, 4, 5};  
cout << *(arr + 4);
```

- a) 1
- b) 3
- c) 5
- d) Compilation error

Q89. Which notation gives an idea of average running time?

- a) Big O
- b) Big Ω
- c) Big Θ
- d) None

Q90. Fill in the blank: A linked list node in C++ is created using the keyword _____.

- a) malloc
- b) new
- c) create
- d) link

Q91. The auxiliary space of an algorithm refers to _____.

- a) Total memory including input
- b) Temporary memory used during execution
- c) Disk space
- d) None of these

Q92. Which of the following best represents logarithmic time complexity?

- a) Binary Search
- b) Linear Search
- c) Bubble Sort
- d) Quick Sort (worst case)

Q93. If an algorithm has time complexity $O(2^n)$, it is classified as _____.

- a) Polynomial time
- b) Logarithmic time
- c) Exponential time
- d) Constant time

Q94. Which operation in a linked list requires $O(n)$ time?

- a) Insert at beginning
- b) Insert at end
- c) Access by index
- d) Both b and c

Q95. What is the best case time complexity of insertion sort?

- a) $O(n)$
- b) $O(n^2)$
- c) $O(\log n)$
- d) $O(1)$

Q96. Fill in the blank: The term "asymptotic analysis" refers to the study of _____.

- a) Input data types
- b) Growth of algorithm with input size
- c) Memory size
- d) Hardware performance

Q97. In which case does Quick Sort perform the best?

- a) When pivot divides array evenly
- b) When pivot is largest element
- c) When array is already sorted
- d) None

Q98. What is the auxiliary space of merge sort?

- a) $O(1)$
- b) $O(\log n)$
- c) $O(n)$
- d) $O(n^2)$

Q99. What is the time complexity of finding the maximum element in an array?

- a) $O(n)$
- b) $O(\log n)$
- c) $O(n^2)$
- d) $O(1)$

Q100. Fill in the blank: In Big O notation, constants and lower-order terms are _____.

- a) Always considered
- b) Ignored
- c) Multiplied
- d) Subtracted

◆ ANSWER KEY - UNIT 1 (Q81-Q100)

- 81. b – Traversing visits each element once $\rightarrow O(n)$.
- 82. b – Merge Sort handles large data efficiently $\rightarrow O(n \log n)$.
- 83. b – Recursive factorial calls n times $\rightarrow O(n)$.
- 84. b – Space complexity = total memory used by algorithm.
- 85. a – Big Ω gives lower bound (best case).
- 86. b – Loop executes n times $\rightarrow O(n)$.
- 87. c – Nested loops $n \times n \rightarrow O(n^2)$.
- 88. c – $*(arr + 4)$ accesses 5th element $\rightarrow 5$.
- 89. c – Big Θ shows average case time.
- 90. b – `new` keyword allocates memory in C++.
- 91. b – Auxiliary space = temporary memory (e.g., recursion stack).
- 92. a – Binary Search has $O(\log n)$ complexity.
- 93. c – $O(2^n)$ = exponential growth, very slow for large inputs.
- 94. d – Insert at end or random access in linked list = $O(n)$.
- 95. a – Best case (already sorted) = $O(n)$.
- 96. b – Asymptotic analysis studies algorithm growth as $n \rightarrow \infty$.
- 97. a – Quick Sort best when pivot divides array evenly.
- 98. c – Merge Sort requires extra $O(n)$ memory.
- 99. a – Must check all n elements $\rightarrow O(n)$.
- 100. b – Constants/lower-order terms ignored in Big O.

Unit -1 Done Thanks :)