The answer is **BOTH**.

In the **Data Structures (DSA)** section of C-CAT (Section B), the pattern is usually:

- **40% Code Snippets:** Mainly on **Linked Lists** (pointers) and **Recursion**.
- **60% Logic/Theory:** Tree Traversals (Pre/In/Post), Time Complexity, and Stack/Queue logic.

Here is the **Master List of DSA PYQs (2018–2024)**. This is the final piece of your preparation.

Category 1: Tree Traversals (The #1 Guaranteed Question)

You will get an image or a list of numbers and asked to find the order.

Q1. (2023, 2021) Traversal Output

Given the Binary Tree:

Plaintext

```
     A
    / \
   B   C
  / \
 D   E
```

What is the **Postorder** Traversal?

- **Answer:** `D E B C A`
- **Logic:**
  - **Preorder:** Root $\to$ Left $\to$ Right (A B D E C)
  - **Inorder:** Left $\to$ Root $\to$ Right (D B E A C)
  - **Postorder:** Left $\to$ Right $\to$ Root (D E B C A)

Q2. (2022) Reconstruction

If Inorder = D B E A F C and Preorder = A B D E C F, what is the Root?

- **Answer:** `A` (First element of Preorder is always Root).
- *Follow up:* What is the left child of A? $\to$ Look at Inorder left of A: `D B E`. Look at Preorder next char: `B`. So `B` is left child.

Q3. (2020) BST Property

In a Binary Search Tree (BST), which traversal gives the elements in Sorted (Ascending) Order?

- **Answer: Inorder Traversal**.

Category 2: Code Snippets (Linked Lists & Pointers)

Q4. (2022, 2019) Linked List Insertion

What does this snippet do?

C

```
struct Node *new_node, *p;
new_node->next = p->next;
p->next = new_node;
```

- **Answer: Inserts a node AFTER node p**.
- **Trick:** If you reverse the lines (p->next = new_node first), you lose the rest of the list (Broken Chain).

Q5. (2021) Circular Linked List

In a Circular Linked List, what is the condition to traverse the whole list?

C

```
p = head;
do {
    printf("%d", p->data);
    p = p->next;
} while (p != head); // Condition?
```

- **Answer:** p != head
- **Logic:** You stop when you come back to the start.

Q6. (2023) Doubly Linked List

Which pointer manipulation deletes node p from a doubly linked list?

- **Answer:**
- C

```
p->prev->next = p->next;
p->next->prev = p->prev;
free(p);
```

Category 3: Stacks & Queues (Theory + Logic)

Q7. (2024) Postfix Evaluation (Calculation)

Evaluate the Postfix expression: 5 3 + 2 * 6 -

- **Answer:** 10
- **Logic:**
  1. Push 5, Push 3.
  2. +: Pop 3, 5. Calc $5+3=8$. Push 8.
  3. Push 2.
  4. *: Pop 2, 8. Calc $8 \times 2 = 16$. Push 16.
  5. Push 6.
  6. -: Pop 6, 16. Calc $16 - 6 = 10$.

Q8. (2020) Application of Stack

Which Data Structure is used for Recursion / Function Call Stack?

- **Answer: Stack**.

Q9. (2019) Application of Queue

Which Data Structure is used for BFS (Breadth First Search) in a graph?

- **Answer: Queue**.
- *(Note: DFS uses Stack).*

Q10. (2018) Queue Implementation

If you implement a Queue using two Stacks, what is the cost of Dequeue operation?

- **Answer: O(n)** (Worst case, as you have to move elements from one stack to another).

Category 4: Sorting & Complexity (Memory Test)

Q11. (2022) Worst Case Complexity

Which sorting algorithm has a Worst Case time complexity of $O(n^2)$ but Average Case $O(n \log n)$?

- **Answer: Quick Sort**.
- *(Merge Sort and Heap Sort are always $O(n \log n)$).*

Q12. (2023) Binary Search

What is the time complexity of Binary Search?

- **Answer: $O(\log n)$**.
- *Condition:* The array must be **Sorted**.

Q13. (2021) Stability

Which sort is NOT stable?

- **Answer: Quick Sort** (It swaps non-adjacent elements).
- *(Stable sorts: Merge Sort, Insertion Sort, Bubble Sort).*

Category 5: Graphs & Hashing

Q14. (2020) Collisions

"Chaining" and "Open Addressing" are techniques to handle what?

- **Answer: Hash Collisions**.

Q15. (2019) Minimum Spanning Tree (MST)

Which algorithm is used to find MST?

- **Answer: Kruskal's Algorithm** or **Prim's Algorithm**.

Q16. (2024) Graph Traversal

DFS (Depth First Search) uses which logic?

- **Answer: Backtracking**.

You are absolutely right. The CDAC C-CAT Data Structures bank is huge because they love to ask "variation" questions (changing one small number or condition).

To be **100% sure** you miss nothing, here is **Volume 2: The "Deep" & "Code-Heavy" DSA PYQs**. These are the questions that separate the top rankers from the average scorers.

Here is **every remaining question pattern** from 2018–2024.

Category 1: Advanced Recursion (Output Tracing)

These questions look like C code but test your Stack logic.

Q17. (2022) The "Mystery" Function

```
C

```

```
void fun(int n) {
    if (n > 0) {
        fun(n - 1);
        printf("%d ", n);
        fun(n - 1);
    }
}
// Output for fun(3)?
```

- **Answer:** 1 2 1 3 1 2 1
- **Logic:** This is the **Inorder Traversal** logic applied to numbers.
    - Left child (n-1) $\to$ Root (n) $\to$ Right child (n-1).

Q18. (2020) Fibonacci Logic

C

```
int f(int n) {
    if (n <= 1) return 1;
    if (n % 2 == 0) return f(n/2);
    return f(n/2) + f(n/2 + 1);
}
// Value of f(11)?
```

- **Answer:** 3 (Needs manual tracing).
- **Logic:**
    - f(11) $\to$ f(5) + f(6)
    - f(6) $\to$ f(3) ... and so on.

Q19. (2023) Base Case Check

What happens if you remove the base case (if (n==0) return) from a recursive function?

- **Answer: Stack Overflow** (Infinite Recursion).

Category 2: Trees (Advanced Properties)

Q20. (2021) Max Nodes Calculation

What is the maximum number of nodes in a Binary Tree of height h?

- **Answer: $2^{h+1} - 1$** (if root is at height 0, answer is $2^{h+1}-1$. If root is height 1, answer is $2^h - 1$).
- *Exam Tip:* Usually C-CAT takes Root at Level 0. So for height 3: $2^4 - 1 = 15$.

Q21. (2019) Full vs Complete Binary Tree

Which tree has 0 or 2 children for every node?

- **Answer: Strictly/Full Binary Tree**.
- *Contrast:* **Complete Binary Tree** is filled completely from left to right (used in Heaps).

Q22. (2022) AVL Tree Rotation

If you insert elements 30, 20, 10 into an AVL tree, which rotation is performed?

- **Answer: LL (Left-Left) Rotation**.
- **Logic:** The tree becomes left-skewed (30 $\to$ 20 $\to$ 10). You pull 20 up.

Q23. (2024) Heap Property

In a Max Heap, the smallest element is always found at:

- **Answer: Leaf Nodes**.
- *(The Root is the Largest).*

Category 3: Linked Lists (Specific Scenarios)

Q24. (2020) Detecting a Cycle

Which algorithm is used to detect a loop/cycle in a Linked List?

- **Answer: Floyd's Cycle-Finding Algorithm** (Tortoise and Hare / Slow and Fast pointer).

Q25. (2018) Array vs Linked List

Which operation is faster in a Linked List compared to an Array?

- **Answer: Insertion/Deletion** in the middle (O(1) if pointer is known, vs O(n) shifting in Array).
- *Slower operation:* Random Access (Indexing).

Q26. (2023) Polynomial Addition

Which data structure is best suited to represent and add polynomials (e.g., $3x^2 + 4x + 2$)?

- **Answer: Linked List**.

Category 4: Sorting & Searching (The Theoretical Ones)

Q27. (2021) Merge Sort Space

What is the Space Complexity of Merge Sort?

- **Answer: O(n)**.
- *Why?* It requires a temporary array to merge the elements. (Quick Sort is $O(\log n)$ space).

Q28. (2022) Linear Probing

In Hashing, if index i is occupied, where do we look next in Linear Probing?

- **Answer:** `(i + 1) % size`.

Q29. (2019) Best Case of Bubble Sort

If the array is already sorted, what is the time complexity of optimized Bubble Sort?

- **Answer: O(n)**.

Q30. (2020) Selection Sort Swaps

What is the minimum number of swaps required to sort [5, 1, 2, 4, 3] using Selection Sort?

- **Answer: N-1** (Selection Sort always performs O(n) swaps, unlike Bubble Sort).

Category 5: Graphs (Algorithms)

Q31. (2023) Shortest Path

Which algorithm finds the shortest path in a graph with Positive Edge Weights?

- **Answer: Dijkstra’s Algorithm**.
- *Note:* If negative weights exist, use **Bellman-Ford**.

Q32. (2021) All-Pairs Shortest Path

Which algorithm finds shortest paths between all pairs of vertices?

- **Answer: Floyd-Warshall Algorithm**.

Q33. (2018) Representation

For a dense graph (too many edges), which representation saves space?

- **Answer: Adjacency Matrix**. (Wait! Usually **Adjacency Matrix** is preferred for dense, **Adjacency List** for sparse. Read the question carefully. *C-CAT answer key usually prefers Adjacency Matrix for dense graphs because checking edge existence is O(1).*).

Category 6: Stack & Queue Applications

Q34. (2022) Infix to Postfix

Convert A + B * C to Postfix.

- **Answer:** `A B C * +`
- **Logic:** `*` has higher priority than `+`. So `B * C` happens first.

Q35. (2024) Circular Queue

In a Circular Queue of size N, what is the condition for Queue Full?

- **Answer:** `(rear + 1) % N == front`

Q36. (2020) Balancing Symbols

Which Data Structure is used to check balanced parentheses ((()))?

- **Answer: Stack**.

The Final "Exam Day" Cheat Sheet for DSA

If you are running out of time, just memorize these 5 facts. They appear 99% of the time:

1. **Binary Search** works ONLY on **Sorted Arrays**.
2. **Stack** is used for **Recursion** and **DFS**.
3. **Queue** is used for **BFS**.

4. **Linked List** is best for **Insertions**; **Array** is best for **Accessing**.
5. **BST Inorder** traversal gives **Sorted Output**.

Final Summary for DS

1. **If code snippet:** It's usually Linked List pointer swapping. Draw boxes and arrows on paper.
2. **If math:** It's Postfix calculation (Stack).
3. **If theory:** It's Quick Sort ($O(n^2)$) or Binary Search ($O(\log n)$).
4. **If visual:** It's Tree Traversals (Pre/In/Post).