---

1. Linked List Questions

These include questions related to linked lists (singly, doubly, circular), their properties, operations, and implementations.

Theory Questions (Linked List)

Beginner Theory

1. **Advantages and disadvantages of linked lists** (2)
2. **Applications of linked lists** (4)
3. **Difference between singly linked lists and doubly linked lists** (1)
4. **How does a doubly linked list differ in traversal compared to a singly linked list** (1)
5. **Applications of doubly linked list** (3)
6. **Advantage of array over linked list** (1)

Intermediate Theory

1. **Memory allocation in linked list** (3)
2. **Memory allocation in array and linked list** (1)
3. **Time complexity of accessing element from linked list when index is known** (1)
4. **Time complexity of inserting element in linked list** (1)
5. **Time complexity of deleting an element from the middle of a linked list** (1)

Advanced Theory

1. **Drawbacks of linked list and recursion** (1)

Practical Questions (Linked List)

Beginner Practical

1. **Convert an array [1, 2, 3, 4, 5] to a linked list** (3)

Intermediate Practical

1. **Find middle of linked list in 1 iteration (fast and slow pointer)** (4)
2. **Get middle element in a linked list** (1)
3. **Linked list implementation** (5)
4. **Singly linked list implementation** (2)
5. **Doubly linked list implementation** (1)
6. **Circular linked list implementation** (1)
7. **Sorted single linked list** (1)
8. **SLL with tail (tail not utilized)** (1)
9. **Append & prepend in linked list** (1)
10. **Insert a node after a node in linked list (check all conditions)** (1)
11. **Function to print all elements of a linked list in order and reverse order** (1)
12. **Remove duplicates from a linked list** (2)
13. **Delete an element from a specific position in singly linked list** (2)
14. **Delete middle element from linked list** (7)
15. **Remove middle element from linked list without knowing the size** (1)
16. **Remove last instance of a value from SLL** (2)
17. **Detect cycle/loop in singly linked list (Floyd's)** (1)
18. **Reverse a singly linked list** (4)
19. **Reverse a doubly linked list** (5)
20. **Insert to a doubly linked list** (1)
21. **Add a node behind the position with particular data in doubly linked list** (1)
22. **Delete front and back node from doubly linked list where data == data** (1)
23. **Traverse a doubly linked list** (1)
24. **Delete function of doubly linked list** (6)
25. **Remove nth node from the end of the list (fast, slow)** (2)

Advanced Practical

1. **Merge two sorted linked lists** (2)
2. **Sort nodes in a linked list** (1)
3. **Remove odd element nodes from linked list** (1)

Summary of Linked List Questions

- **Total Unique Questions**: 37
  - **Theory**: 12 (6 Beginner, 5 Intermediate, 1 Advanced)
  - **Practical**: 25 (1 Beginner, 21 Intermediate, 3 Advanced)
- **Total Appearances**: 75 (sum of counts in parentheses)

---

2. Recursion Questions

These include questions related to recursion concepts, its applications, and recursive implementations.

Theory Questions (Recursion)

Advanced Theory

1. **Space complexity of recursion** (2)
2. **Time complexity in recursion** (1)
3. **Recursion vs loop** (2)
4. **Advantages of recursion** (2)
5. **Disadvantages of recursion** (3)
6. **Applications of recursion** (12)
7. **Direct vs indirect recursion** (10)
8. **Tail vs head recursion** (11)
9. **Binary recursion** (12)
10. **Tail recursion** (4)
11. **Base case in recursion** (1)
12. **Drawbacks of linked list and recursion** (1)

Practical Questions (Recursion)

Beginner Practical

1. **Sum of elements in an array using recursion** (5)

Intermediate Practical

1. **Binary search using recursion** (4)
2. **Fibonacci series using recursion (first 10 elements)** (4)
3. **Factorial using recursion** (2)
4. **Hide "l" from hello using recursion** (1)
5. **Recursively remove a char from string** (6)
6. **Recursion that recurses only 5 times** (1)

Advanced Practical

1. **String permutations** (2)

Summary of Recursion Questions

- **Total Unique Questions**: 20
  - **Theory**: 12 (0 Beginner, 0 Intermediate, 12 Advanced)
  - **Practical**: 8 (1 Beginner, 6 Intermediate, 1 Advanced)
- **Total Appearances**: 83 (sum of counts in parentheses)

---

3. LeetCode Questions

These include questions explicitly referencing LeetCode problems or the "Blind 75" list, as well as general advice to practice on LeetCode.

Practical Questions (LeetCode)

Advanced Practical

1. **Two sum (LeetCode)** (1)
2. **Valid parentheses (LeetCode)** (2)
3. **Buy and sell stock (sliding window approach)** (2)
4. **Practice problems from blind 75 LeetCode, learn optimal solutions** (5)
5. **Solve more problems on platforms like LeetCode, HackerRank, GeeksforGeeks (application-level programs)** (1)

Summary of LeetCode Questions

- **Total Unique Questions**: 5
  - **Theory**: 0
  - **Practical**: 5 (0 Beginner, 0 Intermediate, 5 Advanced)
- **Total Appearances**: 11 (sum of counts in parentheses)

Overall Summary

- **Linked List Questions**:
  - Unique: 37 (12 Theory, 25 Practical)
  - Appearances: 75
- **Recursion Questions**:
  - Unique: 20 (12 Theory, 8 Practical)
  - Appearances: 83
- **LeetCode Questions**:
  - Unique: 5 (0 Theory, 5 Practical)
  - Appearances: 11
- **Total Unique Questions Across Categories**: 62
- **Total Appearances Across Categories**: 169

Notes

- **Categorization**: Questions were grouped strictly based on their relevance to linked lists, recursion, or LeetCode. For example, "Drawbacks of linked list and recursion" appears in both Linked List and Recursion categories due to its dual relevance.
- **Duplicates**: Consolidated duplicates within each category, with appearance counts in parentheses.
- **Exclusions**: General advice like "practice more logical workouts" was excluded unless explicitly tied to LeetCode or a specific problem.

- **Level Assignment**: Questions were assigned to beginner, intermediate, or advanced based on complexity (e.g., basic linked list operations are intermediate, while LeetCode Blind 75 problems are advanced).

- **Visual Representation**: If you'd like a chart (e.g., bar graph of question counts per category or level), let me know, and I can describe how to visualize it or provide a textual representation.