DSA

Linked list Questions

Easy :

21: Merge Two Sorted Lists

203: Remove Linked List Elements

206: Reverse Linked List

876: Middle of the Linked List

Medium :

19: Remove Nth Node From End of List

24: Swap Nodes in Pairs

116: Populating Next Right Pointers in Each Node

117. Populating Next Right Pointers in Each Node II

237: Delete Node in a Linked List

707: Design Linked List

Hard :

23: Merge k Sorted Lists

25: Reverse Nodes in k-Group

Abdul Bari

- http://www.youtube.com/playlist?list=PLDN4rrl48XKpZkf03iYFl-O29szjTrs_O
- This channel has a playlist called “Algorithms” that covers a wide range of data structures and algorithms.

Telusko

- http://www.youtube.com/playlist?list=PLsyeobzWxl7oRKwDi7wjrANsbhTX0IK0J)
- This channel has a playlist called “Data Structures using Java” that teaches you how to implement common data structures in Java.

freeCodeCamp.org

- http://www.youtube.com/playlist?list=PLWKjhJtqVAbnqBxcdjVGgT3uVR10bzTEB
- This channel has a playlist called “Python Tutorials” that includes some videos on data structures and algorithms, although it is primarily focused on Python.

Caleb Curry

- http://www.youtube.com/playlist?list=PL_c9BZzLwBRLpDEpYRFXKBN-2ZCsAx0ps
- This channel has a playlist called “Data Structures and Algorithms” that covers a variety of data structures and algorithms in a clear and concise way.

mycodeschool

- http://www.youtube.com/playlist?list=PL2_aWCzGMAwLZp6LMUKI3cc7pgGsasm2_
- This channel has a playlist called “Pointers in C/C++” that covers pointers, which are an important concept for understanding data structures.

1. Array-Based Questions:
   a. Find the maximum subarray sum.
   b. Rotate an array.
   c. Implement an algorithm to find duplicates in an array.
   d. Merge two sorted arrays.
2. Linked List Questions:
   a. Reverse A Linked List.
   b. Detect A Cycle In A Linked List.

c. Find The Intersection Point Of Two Linked Lists.
d. Implement In-Place Reversal Of A Linked List.

3. Tree-Based Questions:
   a. Binary Tree Traversal (In-Order, Pre-Order, Post-Order).
   b. Find The Height (Depth) Of A Binary Tree.
   c. Check If A Binary Tree Is Balanced.
   d. Find The Lowest Common Ancestor In A Binary Tree.

4. Graph-Based Questions:
   a. Depth-First Search (DFS) And Breadth-First Search (BFS) Algorithms.
   b. Check If A Graph Is Cyclic.
   c. Find The Shortest Path In An Unweighted Graph.
   d. Topological Sort Of A Directed Acyclic Graph (DAG).

5. Sorting And Searching:
   a. Implement Binary Search.
   b. Implement Quicksort Or Mergesort.
   c. Find The Kth Smallest/Largest Element In An Array.
   d. Search In A Rotated Sorted Array.

6. Dynamic Programming:
   a. Longest Increasing Subsequence.
   b. 0/1 Knapsack Problem.
   c. Coin Change Problem.
   d. Edit Distance Between Two Strings.

7. Miscellaneous:
   a. Implement A Stack And A Queue.
   b. Design And Implement A Cache.
   c. Find The Majority Element In An Array

3 things to focus for Technical interview !

1. Basic Programming Concepts:

- Variables and Data Types

- Control Structures (if statements, loops)

- Functions and Methods

- Object-Oriented Programming (OOP) Principles

- Error Handling and Exceptional Control Flow

- File I/O and Data Serialization

- Modularization and Code Reusability

2. Algorithms:

- Sorting Algorithms (e.g., Bubble Sort, Quick Sort)

- Searching Algorithms (e.g., Binary Search)

- Recursion

- Time and Space Complexity Analysis

- Greedy Algorithms

- Dynamic Programming

- Backtracking Algorithms

- Divide and Conquer Algorithms

- Randomized Algorithms

3. Data Structures:

- Arrays and Linked Lists

- Stacks and Queues

- Trees (Binary Trees, Binary Search Trees)

- Hash Tables

- Graphs and Graph Algorithms

- Heaps and Priority Queues

- Disjoint Set Union (Union-Find)

- Tries (Prefix Trees)

- Segment Trees

- Fenwick Trees (Binary Indexed Trees)

4. Additional Concepts:

- Pointers and Memory Management
- Exception Handling
- Concurrency and Parallelism
- Networking and Socket Programming
- Cryptography and Security Principles
- Web Development Principles
- Database Concepts and SQL
- Regular Expressions
- Software Design Patterns
- Testing and Debugging Practices

1. Array-Based Questions:
   a. Find the maximum subarray sum.
   b. Rotate an array.
   c. Implement an algorithm to find duplicates in an array.
   d. Merge two sorted arrays.
2. Linked List Questions:
   a. Reverse A Linked List.
   b. Detect A Cycle In A Linked List.
   c. Find The Intersection Point Of Two Linked Lists.
   d. Implement In-Place Reversal Of A Linked List.
3. Tree-Based Questions:
   a. Binary Tree Traversal (In-Order, Pre-Order, Post-Order
   b. Find The Height (Depth) Of A Binary Tree.
   c. Check If A Binary Tree Is Balanced.
   d. Find The Lowest Common Ancestor In A Binary Tree.
4. Graph-Based Questions:
   a. Depth-First Search (DFS) And Breadth-First Search (Bl
   b. Check If A Graph Is Cyclic.

c. Find The Shortest Path In An Unweighted Graph.
d. Topological Sort Of A Directed Acyclic Graph (DAG).

5. Sorting And Searching:
   a. Implement Binary Search.
   b. Implement Quicksort Or Mergesort.
   c. Find The Kth Smallest/Largest Element In An Array.
   d. Search In A Rotated Sorted Array.

6. Dynamic Programming:
   a. Longest Increasing Subsequence.
   b. 0/1 Knapsack Problem.
   c. Coin Change Problem.
   d. Edit Distance Between Two Strings.

7. Miscellaneous:
   a. Implement A Stack And A Queue.
   b. Design And Implement A Cache.
   c. Find The Majority Element In An Array.