70 DSA Questions

Arrays & Numbers

1. Find the maximum and minimum element in an array.
2. Find the second largest element in an array.
3. Rotate an array by k positions.
4. Find the intersection of two arrays.
5. Find the union of two arrays.
6. Find the majority element (appears > n/2 times).
7. Count pairs with a given sum.
8. Rearrange positive and negative numbers alternately.
9. Sort an array of 0s, 1s, and 2s without extra space.
10. Merge overlapping intervals.
11. Find the subarray with a given sum.
12. Find the smallest missing positive number.
13. Find the product of array elements except self.
14. Find the minimum number of jumps to reach array end.
15. Find leaders in an array.

Strings

16. Count vowels and consonants in a string.
17. Remove duplicates from a string.
18. Check if two strings are anagrams.
19. Find the first non-repeating character in a string.

20. Compress a string (e.g., “aaabb” → “a3b2”).

21. Check if two strings are rotations of each other.

22. Implement strStr() (find substring index).

23. Find all permutations of a string.

24. Find the longest common prefix among strings.

25. Group words that are anagrams.

Linked List

26. Reverse a linked list.

27. Detect a cycle in a linked list.

28. Find the middle of a linked list.

29. Merge two sorted linked lists.

30. Remove the Nth node from the end.

31. Add two numbers represented by linked lists.

32. Check if a linked list is a palindrome.

33. Remove duplicates from a sorted linked list.

34. Find intersection point of two linked lists.

35. Flatten a linked list.

Stack & Queue

36. Implement a stack using queues.

37. Implement a queue using stacks.

38. Evaluate a postfix expression.

39. Find the next greater element for each element.

40. Implement a min stack (support getMin in O(1)).

41. Check for balanced parentheses.

42. Implement an LRU cache.

43. Implement a circular queue.

44. Sort a stack using recursion.

45. Decode a string with nested patterns (e.g., “3[a2[b]]”).

Binary Tree / BST

46. Perform inorder, preorder, and postorder traversal.

47. Find the height of a binary tree.

48. Check if two trees are identical.

49. Check if a binary tree is balanced.

50. Find the lowest common ancestor (LCA).

51. Find the diameter of a binary tree.

52. Convert a binary tree to its mirror.

53. Find the top view of a binary tree.

54. Level order traversal of a tree.

55. Check if a binary tree is a BST.

Graphs

56. Implement BFS and DFS.

57. Detect a cycle in an undirected graph.

58. Detect a cycle in a directed graph.

59. Find shortest path in an unweighted graph (BFS).

60. Find the number of connected components.

61. Implement Dijkstra's algorithm.

62. Implement Topological Sort.

63. Implement Kruskal's algorithm (MST).

64. Implement Prim's algorithm (MST).

65. Find if a path exists between two nodes.

Dynamic Programming & Recursion

66. Compute Fibonacci using memoization.

67. Solve 0/1 Knapsack problem.

68. Find the longest increasing subsequence (LIS).

69. Find the longest common subsequence (LCS).

70. Find the minimum edit distance between two strings.

Project Ideas

AI Content Generation Platform

Description: A web app that generates marketing content, blog posts, or social media captions using AI prompts.

AI Integration: GPT-based text generation, sentiment adjustment, tone control.

Tech Stack:

- Frontend: Next.js, TailwindCSS

- Backend: FastAPI or Node.js (Express)
- Database: PostgreSQL or MongoDB
- AI: OpenAI API or Anthropic Claude
- Extras: Redis for caching, AWS S3 for media

AI Resume Analyzer & Job Matcher

Description: Users upload resumes; the app parses and matches them with job descriptions using NLP.

AI Integration: Resume parsing, skill extraction, job match scoring using embeddings.

Tech Stack:

- Frontend: React + TypeScript
- Backend: Python (FastAPI)
- Database: PostgreSQL
- AI: OpenAI Embeddings + LangChain
- Extras: Elasticsearch for job search

Intelligent Code Review Dashboard

Description: A platform where developers upload code, and the AI gives code quality suggestions.

AI Integration: LLM-based static code analysis, bug suggestions, and code explanations.

Tech Stack:

- Frontend: Next.js
- Backend: Node.js (Express) or Django
- Database: MongoDB

- AI: OpenAI Code Interpreter API or Code Llama
- Extras: GitHub OAuth integration

AI-Powered Customer Support Chatbot

Description: Custom chatbot for businesses that learns from FAQ and customer history.

AI Integration: Conversational AI, retrieval-augmented generation (RAG) using company docs.

Tech Stack:

- Frontend: React + TailwindCSS
- Backend: FastAPI
- Database: PostgreSQL + Pinecone (vector DB)
- AI: LangChain + OpenAI or Claude API
- Extras: WebSocket for live chat

Smart Health Tracker Dashboard

Description: Users upload wearable data; AI predicts anomalies and provides health insights.

AI Integration: Time-series anomaly detection, predictive analytics using ML models.

Tech Stack:

- Frontend: React + Chart.js
- Backend: Flask or FastAPI
- Database: MongoDB
- AI: scikit-learn / PyTorch for predictive model
- Extras: Kafka for streaming data

AI-Based E-Learning Platform

Description: A personalized course platform that recommends learning paths and quizzes.

AI Integration: Recommendation engine, adaptive difficulty quiz generator.

Tech Stack:

- Frontend: Next.js
- Backend: Node.js (NestJS)
- Database: PostgreSQL
- AI: TensorFlow Recommenders + OpenAI GPT
- Extras: Supabase Auth or Firebase

AI Marketing Insights Dashboard

Description: Analyze campaign data and generate automatic insights or performance summaries.

AI Integration: NLP-based report summarization, predictive trend analysis.

Tech Stack:

- Frontend: React + D3.js
- Backend: Python (FastAPI)
- Database: PostgreSQL
- AI: GPT + Prophet for forecasting
- Extras: Kafka for real-time metrics

Voice-Powered Task Manager

Description: A productivity app where users can manage tasks using speech commands.

AI Integration: Speech-to-text, intent recognition, and task automation.

Tech Stack:

- Frontend: React Native or React
- Backend: Node.js
- Database: Firebase
- AI: Whisper API (for STT), GPT for task interpretation
- Extras: Push notifications via FCM

AI-Driven Financial Expense Tracker

Description: Tracks expenses and categorizes them automatically using AI.

AI Integration: Transaction classification using NLP and anomaly detection.

Tech Stack:

- Frontend: Vue.js
- Backend: Django REST Framework
- Database: PostgreSQL
- AI: scikit-learn or OpenAI API for text classification
- Extras: Stripe API for transaction import

AI Recruitment Assistant

Description: Automates candidate screening, interview scheduling, and feedback summarization.

AI Integration: Resume ranking, email drafting, and interview summary generation.

Tech Stack:

- Frontend: Next.js

- Backend: FastAPI
- Database: MongoDB
- AI: OpenAI GPT + LangChain
- Extras: Google Calendar API integration