The most commonly asked algorithms in front-end and back-end interviews:

Front-end Interviews:

1. Palindrome Check
2. Anagram Detection
3. Fibonacci Series
4. Reverse a String
5. Fizz Buzz
6. Implement Debouncing and Throttling
7. Implement Event Delegation
8. Implement Closure
9. Implement Promise
10. Implement Currying
11. Implement Memoization
12. Implement Throttling
13. Implement Debouncing
14. Implement Merge Sort
15. Implement Quick Sort
16. Implement Binary Search
17. Implement Tree Traversal (DFS, BFS)
18. Implement Linked List Operations
19. Implement Stack and Queue
20. Implement Trie
21. Implement DOM Manipulation
22. Implement Event Handling
23. Implement Array Methods (map, filter, reduce, etc.)

24. Implement Asynchronous Programming (callbacks, promises, async/await)
25. Implement Templating and String Manipulation
26. Implement Form Validation
27. Implement Responsive Design
28. Implement CSS Positioning and Layout
29. Implement Performance Optimization Techniques
30. Implement Accessibility Standards

Back-end Interviews:

1. Linked List Operations
2. Binary Search
3. Breadth-First Search (BFS)
4. Depth-First Search (DFS)
5. Merge Sort
6. Quick Sort
7. Implement LRU Cache
8. Implement Rate Limiting
9. Implement Caching
10. Implement Message Queuing
11. Implement Hash Table
12. Implement Graph Algorithms (Dijkstra, Kruskal, Prim)
13. Implement Heap (Min/Max)
14. Implement Trie
15. Implement Bloom Filter
16. Implement Sliding Window Technique
17. Implement Knapsack Problem
18. Implement Reservoir Sampling

19. Implement Topological Sort
20. Implement Longest Common Subsequence (LCS)
21. Implement Concurrency Control (Locks, Semaphores, Mutex)
22. Implement Database Normalization
23. Implement Microservices Architecture
24. Implement Server-Side Caching
25. Implement Load Balancing
26. Implement Websockets and Real-Time Communication
27. Implement Secure Authentication and Authorization
28. Implement API Design and RESTful Principles
29. Implement Logging and Monitoring
30. Implement Containerization and Orchestration (Docker, Kubernetes)

The most commonly asked design system-related topics in front-end and back-end interviews:

Front-end Interviews:

1. Atomic Design Principles
2. Design Token Management
3. UI Component Library Development
4. Responsive and Adaptive Design
5. Accessibility and Inclusive Design
6. Design System Versioning and Governance
7. Design System Documentation and Onboarding
8. Design System Testing and Quality Assurance
9. Design System Performance Optimization
10. Design System Theming and Customization

11. Design System Styling Approaches (CSS-in-JS, CSS Modules, etc.)
12. Design System Component Composition and Reuse
13. Design System State Management and Lifecycle
14. Design System Internationalization and Localization
15. Design System Accessibility Integration
16. Design System Performance Budgeting
17. Design System Design Principles (Consistency, Modularity, Scalability)
18. Design System Design System Tooling (Figma, Sketch, Storybook, etc.)
19. Design System Icon and Asset Management
20. Design System Interaction and Microinteractions
21. Design System Typography and Text Styling
22. Design System Color and Palette Management
23. Design System Spacing and Layout Systems
24. Design System Animation and Motion Design
25. Design System Responsive Grid and Layout
26. Design System Form and Input Controls
27. Design System Data Visualization and Charts
28. Design System Content Strategy and Messaging
29. Design System Usability and User Testing
30. Design System Maintenance and Adoption

Back-end Interviews:

1. Design System API Development
2. Design System Data Model and Schema
3. Design System Content Management
4. Design System Deployment and Distribution
5. Design System Monitoring and Analytics
6. Design System Dependency Management
7. Design System Security and Compliance

8. Design System Integrations with Other Systems

9. Design System Scalability and Extensibility

10. Design System Architectural Patterns (Monorepo, Modular, Federated)

11. Design System Headless and Decoupled Architectures

12. Design System API Design and Versioning

13. Design System Caching and Performance Optimization

14. Design System Logging and Observability

15. Design System Continuous Integration and Deployment

16. Design System Microservices and Service Mesh

17. Design System Authentication and Authorization

18. Design System Event-Driven and Serverless Architectures

19. Design System Offline and Edge Computing Support

20. Design System Internationalization and Localization Infrastructure

21. Design System Deployment Automation and Infrastructure as Code

22. Design System Monitoring and Alerting

23. Design System Incident Response and Disaster Recovery

24. Design System Rollback and Rollforward Strategies

25. Design System Change Management and Feature Flagging

The most commonly asked design patterns in front-end and back-end interviews:

Front-end Interviews:

1. Singleton

2. Factory

3. Observer

4. Decorator

5. Adapter

6. Facade
7. Composite
8. Strategy
9. Command
10. Mediator
11. Memento
12. Prototype
13. Builder
14. Flyweight
15. Iterator
16. State
17. Template Method
18. Visitor
19. Proxy
20. Chain of Responsibility
21. Pub/Sub
22. MVC (Model-View-Controller)
23. MVP (Model-View-Presenter)
24. MVVM (Model-View-ViewModel)
25. Flux
26. Redux
27. React Context
28. Higher-Order Components
29. Render Props
30. Hooks
31. Functional Programming
32. Reactive Programming
33. Immutable Data
34. Dependency Injection

35. Micro-Frontends

36. Web Components

37. Service Worker

38. Progressive Web Apps

39. Responsive Design

40. Atomic Design

Back-end Interviews:

1. Singleton

2. Factory

3. Adapter

4. Facade

5. Strategy

6. Template Method

7. Proxy

8. Chain of Responsibility

9. Decorator

10. Observer

11. Command

12. Memento

13. Visitor

14. Composite

15. Flyweight

16. Interpreter

17. Iterator

18. Mediator

19. State

20. MVC (Model-View-Controller)

21. MVP (Model-View-Presenter)

22. MVVM (Model-View-ViewModel)

23. Layered Architecture

24. Repository

25. Unit of Work

26. Service

27. Data Mapper

28. Identity Map

29. Dependency Injection

30. Inversion of Control

31. Event-Driven Architecture

32. Message Queue

33. Publish-Subscribe

34. Circuit Breaker

35. Retry Pattern

36. Bulkhead

37. Saga

38. CQRS (Command Query Responsibility Segregation)

39. Event Sourcing

40. Domain-Driven Design

The most commonly asked data structures in front-end and back-end interviews:

Front-end Interviews:

1. Array

2. Linked List

3. Stack

4. Queue

5. Hash Table

6. Set

7. Map

8. Tree

9. Binary Tree

10. Binary Search Tree

11. Trie

12. Heap

13. Graph

14. Doubly Linked List

15. Circular Linked List

16. Priority Queue

17. Deque

18. Disjoint Set

19. Bloom Filter

20. Segment Tree

21. Fenwick Tree

22. Suffix Array

23. Radix Tree

24. Quadtree

25. Octree

26. Interval Tree

27. Kd-Tree

28. R-Tree

29. Directed Acyclic Graph (DAG)

30. Adjacency List

31. Adjacency Matrix

32. Directed Graph

33. Undirected Graph

34. Weighted Graph

35. Unweighted Graph

36. Bipartite Graph

37. Planar Graph

38. Topological Sort

39. Strongly Connected Components

40. Minimum Spanning Tree

Back-end Interviews:

1. Array

2. Linked List

3. Stack

4. Queue

5. Hash Table

6. Set

7. Map

8. Tree

9. Binary Tree

10. Binary Search Tree

11. Trie

12. Heap

13. Graph

14. Doubly Linked List

15. Circular Linked List

16. Priority Queue

17. Deque

18. Disjoint Set

19. Bloom Filter

20. Segment Tree

21. Fenwick Tree

22. Suffix Array

23. Radix Tree

24. Quadtree

25. Octree

26. Interval Tree

27. Kd-Tree

28. R-Tree

29. Directed Acyclic Graph (DAG)

30. Adjacency List

31. Adjacency Matrix

32. Directed Graph

33. Undirected Graph

34. Weighted Graph

35. Unweighted Graph

36. Bipartite Graph

37. Planar Graph

38. Topological Sort

39. Strongly Connected Components

40. Minimum Spanning Tree

The most commonly asked SOLID principles in front-end and back-end interviews:

Front-end Interviews:

1. Single Responsibility Principle

2. Open/Closed Principle

3. Liskov Substitution Principle

4. Interface Segregation Principle
5. Dependency Inversion Principle
6. Composition over Inheritance
7. Don't Repeat Yourself (DRY)
8. Separation of Concerns
9. Loose Coupling
10. High Cohesion
11. Encapsulation
12. Abstraction
13. Modularity
14. Information Hiding
15. Polymorphism
16. Immutability
17. Functional Programming
18. Pure Functions

19. Idempotency
20. Declarative Programming
21. Reactive Programming
22. Observability
23. Testability
24. Maintainability
25. Extensibility
26. Scalability
27. Performance
28. Accessibility
29. Internationalization
30. Separation of Layout and Logic
31. Unidirectional Data Flow
32. Immutable State

33. Stateless Components

34. Container and Presentational Components

35. Higher-Order Components

36. React Hooks

37. Prop Drilling

38. Context API

39. Web Components

40. Shadow DOM

Back-end Interviews:

1. Single Responsibility Principle

2. Open/Closed Principle

3. Liskov Substitution Principle

4. Interface Segregation Principle

5. Dependency Inversion Principle

6. Composition over Inheritance

7. Don't Repeat Yourself (DRY)

8. Separation of Concerns

9. Loose Coupling

10. High Cohesion

11. Encapsulation

12. Abstraction

13. Modularity

14. Information Hiding

15. Polymorphism

16. Immutability

17. Functional Programming

18. Pure Functions

19. Idempotency

20. Declarative Programming
21. Event-Driven Programming
22. Observability
23. Testability
24. Maintainability
25. Extensibility
26. Scalability
27. Performance
28. Fault Tolerance
29. Resilience
30. Asynchronous Programming
31. Concurrency
32. Parallelism
33. Message Queues
34. Microservices
35. Domain-Driven Design
36. Hexagonal Architecture
37. Onion Architecture
38. Clean Architecture
39. CQRS (Command Query Responsibility Segregation)
40. Event Sourcing

The most commonly asked coding challenges in front-end and back-end interviews:

Front-end Interviews:

1. FizzBuzz
2. Palindrome Check

3. Reverse a String

4. Anagram Check

5. Two Sum

6. Merge Two Sorted Arrays

7. Implement a Stack

8. Implement a Queue

9. Implement a Linked List

10. Implement a Binary Search Tree

11. Implement a Hash Table

12. Implement a Trie

13. Implement a Heap

14. Traverse a Binary Tree

15. Depth-First Search (DFS)

16. Breadth-First Search (BFS)

17. Implement Debouncing

18. Implement Throttling

19. Implement a Carousel

20. Implement a Modal

21. Implement a Dropdown

22. Implement a Tooltip

23. Implement a Pagination

24. Implement a Search Autocomplete

25. Implement a Infinite Scroll

26. Implement a Lazy Loading

27. Implement a Responsive Layout

28. Implement a Drag and Drop

29. Implement a State Management System

30. Implement a Virtual Scrolling

31. Implement a Webworkers

32. Implement a Service Worker

33. Implement a Progressive Web App

34. Implement a Single Page Application

35. Implement a Routing System

36. Implement a Form Validation

37. Implement a Accessibility Features

38. Implement a Code Splitting

39. Implement a Server-Side Rendering

40. Implement a Internationalization

Back-end Interviews:

1. FizzBuzz

2. Palindrome Check

3. Reverse a String

4. Anagram Check

5. Two Sum

6. Merge Two Sorted Arrays

7. Implement a Stack

8. Implement a Queue

9. Implement a Linked List

10. Implement a Binary Search Tree

11. Implement a Hash Table

12. Implement a Trie

13. Implement a Heap

14. Traverse a Binary Tree

15. Depth-First Search (DFS)

16. Breadth-First Search (BFS)

17. Implement a Web Server

18. Implement a RESTful API

19. Implement a CRUD Operation
20. Implement a Authentication System
21. Implement a Authorization System
22. Implement a Logging System
23. Implement a Monitoring System
24. Implement a Caching System
25. Implement a Message Queuing System
26. Implement a Pub/Sub System
27. Implement a Microservices Architecture
28. Implement a Load Balancing
29. Implement a Database Connection Pool
30. Implement a Database Migration
31. Implement a Database Indexing
32. Implement a Transaction Management
33. Implement a Exception Handling
34. Implement a Asynchronous Programming
35. Implement a Concurrency Control
36. Implement a Deadlock Detection
37. Implement a Rate Limiting
38. Implement a Distributed Tracing
39. Implement a Containerization
40. Implement a Continuous Integration/Deployment

The most commonly asked live coding challenges in front-end and back-end interviews:

Front-end Interviews:

1. Implement a Counter

2. Implement a Todo List

3. Implement a Stopwatch

4. Implement a Tic Tac Toe Game

5. Implement a Snake Game

6. Implement a Tetris Game

7. Implement a Flappy Bird Game

8. Implement a Memory Game

9. Implement a Sudoku Solver

10. Implement a Chess Board

11. Implement a Slider/Carousel

12. Implement a Modal/Popup

13. Implement a Dropdown/Select

14. Implement a Accordion

15. Implement a Tabs

16. Implement a Infinite Scroll

17. Implement a Search Autocomplete

18. Implement a Pagination

19. Implement a Form Validation

20. Implement a Drag and Drop

21. Implement a Sortable List

22. Implement a Image Gallery

23. Implement a Video Player

24. Implement a Chart/Graph

25. Implement a Data Table

26. Implement a Calendar

27. Implement a Notification System

28. Implement a Chat Application

29. Implement a Polling Application

30. Implement a Real-time Collaboration

31. Implement a Progressive Web App
32. Implement a Server-Side Rendering
33. Implement a Code Splitting
34. Implement a Lazy Loading
35. Implement a Responsive Layout
36. Implement a Accessibility Features
37. Implement a State Management
38. Implement a Unit Tests
39. Implement a End-to-End Tests
40. Implement a Performance Optimization

Back-end Interviews:

1. Implement a RESTful API
2. Implement a CRUD Operation
3. Implement a Authentication System
4. Implement a Authorization System
5. Implement a Logging System
6. Implement a Monitoring System
7. Implement a Caching System
8. Implement a Message Queuing System
9. Implement a Pub/Sub System
10. Implement a Microservices Architecture
11. Implement a Load Balancing
12. Implement a Database Connection Pool
13. Implement a Database Migration
14. Implement a Database Indexing
15. Implement a Transaction Management
16. Implement a Exception Handling
17. Implement a Asynchronous Programming

18. Implement a Concurrency Control
19. Implement a Deadlock Detection
20. Implement a Rate Limiting
21. Implement a Distributed Tracing
22. Implement a Containerization
23. Implement a Continuous Integration/Deployment
24. Implement a Web Server
25. Implement a Socket.IO Server
26. Implement a WebSocket Server
27. Implement a Kafka Producer/Consumer
28. Implement a Redis Cache
29. Implement a MongoDB CRUD
30. Implement a PostgreSQL CRUD
31. Implement a Serverless Function
32. Implement a Serverless Event Handling
33. Implement a GraphQL API
34. Implement a gRPC API
35. Implement a Webscraper
36. Implement a Batch Processing
37. Implement a Streaming Processing
38. Implement a Data Pipelines
39. Implement a CI/CD Pipelines
40. Implement a Monitoring and Alerting

The most commonly asked tasks in front-end and back-end interviews:

Front-end Interviews:

1. Reverse a string

2. Implement a palindrome checker
3. Implement a fizzbuzz solution
4. Implement a two-sum solution
5. Implement a merge sorted arrays
6. Implement a stack data structure
7. Implement a queue data structure
8. Implement a linked list
9. Implement a binary search tree
10. Implement a hash table
11. Implement a trie data structure
12. Implement a heap data structure
13. Implement a depth-first search
14. Implement a breadth-first search
15. Implement a debouncing function
16. Implement a throttling function
17. Implement a carousel slider
18. Implement a modal/dialog
19. Implement a dropdown menu
20. Implement a tooltip
21. Implement a pagination component
22. Implement a search autocomplete
23. Implement an infinite scroll
24. Implement a lazy loading
25. Implement a responsive layout
26. Implement a drag and drop
27. Implement a state management system
28. Implement a virtual scrolling
29. Implement a web worker
30. Implement a service worker

31. Implement a progressive web app
32. Implement a single page application
33. Implement a routing system
34. Implement a form validation
35. Implement accessibility features
36. Implement code splitting
37. Implement server-side rendering
38. Implement internationalization

Back-end Interviews:

1. Implement a RESTful API
2. Implement a CRUD operation
3. Implement an authentication system
4. Implement an authorization system
5. Implement a logging system
6. Implement a monitoring system
7. Implement a caching system
8. Implement a message queuing system
9. Implement a pub/sub system
10. Implement a microservices architecture
11. Implement a load balancing solution
12. Implement a database connection pool
13. Implement a database migration
14. Implement database indexing
15. Implement a transaction management
16. Implement exception handling
17. Implement asynchronous programming
18. Implement concurrency control
19. Implement deadlock detection

20. Implement rate limiting
21. Implement distributed tracing
22. Implement containerization
23. Implement continuous integration/deployment
24. Implement a web server
25. Implement a socket.io server
26. Implement a websocket server
27. Implement a Kafka producer/consumer
28. Implement a Redis cache
29. Implement a MongoDB CRUD
30. Implement a PostgreSQL CRUD
31. Implement a serverless function
32. Implement a serverless event handling
33. Implement a GraphQL API
34. Implement a gRPC API
35. Implement a web scraper
36. Implement batch processing
37. Implement streaming processing
38. Implement data pipelines
39. Implement CI/CD pipelines
40. Implement monitoring and alerting

The most commonly asked coding challenges in front-end and back-end interviews:

Front-end Interviews:

1. Implement a Counter
2. Implement a Todo List

3. Implement a Stopwatch

4. Implement a Tic Tac Toe Game

5. Implement a Snake Game

6. Implement a Tetris Game

7. Implement a Flappy Bird Game

8. Implement a Memory Game

9. Implement a Sudoku Solver

10. Implement a Chess Board

11. Implement a Slider/Carousel

12. Implement a Modal/Popup

13. Implement a Dropdown/Select

14. Implement an Accordion

15. Implement Tabs

16. Implement Infinite Scroll

17. Implement Search Autocomplete

18. Implement Pagination

19. Implement Form Validation

20. Implement Drag and Drop

21. Implement Sortable List

22. Implement Image Gallery

23. Implement Video Player

24. Implement Chart/Graph

25. Implement Data Table

26. Implement Calendar

27. Implement Notification System

28. Implement Chat Application

29. Implement Polling Application

30. Implement Real-time Collaboration

31. Implement Progressive Web App

32. Implement Server-Side Rendering
33. Implement Code Splitting
34. Implement Lazy Loading
35. Implement Responsive Layout
36. Implement Accessibility Features
37. Implement State Management
38. Implement Unit Tests
39. Implement End-to-End Tests
40. Implement Performance Optimization

Back-end Interviews:

1. Implement a RESTful API
2. Implement a CRUD Operation
3. Implement Authentication System
4. Implement Authorization System
5. Implement Logging System
6. Implement Monitoring System
7. Implement Caching System
8. Implement Message Queuing System
9. Implement Pub/Sub System
10. Implement Microservices Architecture
11. Implement Load Balancing
12. Implement Database Connection Pool
13. Implement Database Migration
14. Implement Database Indexing
15. Implement Transaction Management
16. Implement Exception Handling
17. Implement Asynchronous Programming
18. Implement Concurrency Control

19. Implement Deadlock Detection
20. Implement Rate Limiting
21. Implement Distributed Tracing
22. Implement Containerization
23. Implement Continuous Integration/Deployment
24. Implement Web Server
25. Implement Socket.IO Server
26. Implement WebSocket Server
27. Implement Kafka Producer/Consumer
28. Implement Redis Cache
29. Implement MongoDB CRUD
30. Implement PostgreSQL CRUD
31. Implement Serverless Function
32. Implement Serverless Event Handling
33. Implement GraphQL API
34. Implement gRPC API
35. Implement Web Scraper
36. Implement Batch Processing
37. Implement Streaming Processing
38. Implement Data Pipelines
39. Implement CI/CD Pipelines
40. Implement Monitoring and Alerting


S
SOHEIB
@soheibkiani