Complete DSA Patterns Guide (Easy → Hard)

EVERY Pattern Covered

1. Array & String Patterns

Easy Level

- **Linear Scan/Traversal** - Simple iteration through arrays
- **Two Pointers (Same Direction)** - Both pointers move forward
- **Frequency Counter** - Using hash map to count occurrences
- **Prefix Sum** - Pre-compute cumulative sums
- **Running Sum** - Calculate cumulative sum on the fly
- **In-place Modification** - Modify array without extra space

Medium Level

- **Two Pointers (Opposite Direction)** - Start from both ends
- **Sliding Window (Fixed Size)** - Fixed window moving through array
- **Sliding Window (Variable Size)** - Window expands/contracts based on condition
- **Kadane's Algorithm** - Maximum subarray sum
- **Dutch National Flag** - Three-way partitioning
- **Cyclic Sort** - For arrays with numbers in range [1, n]
- **Fast & Slow Pointers (Array)** - Find duplicates in array
- **Suffix Array** - String pattern matching

Hard Level

- **Monotonic Stack** - Stack maintaining order property
- **Next Greater/Smaller Element** - Using stack for next element problems
- **Rain Water Trapping** - Calculate trapped water patterns
- **Stock Buy/Sell (Multiple Transactions)** - Complex DP on arrays
- **Boyer-Moore Voting Algorithm** - Majority element finding
- **Reservoir Sampling** - Random sampling from stream
- **Merge Overlapping Ranges** - Complex interval merging

---

2. Hash Map/Set Patterns

Easy Level

- **Direct Mapping** - Map keys to values directly
- **Set for Duplicates** - Check existence using sets
- **Counting Frequency** - Count occurrences of elements
- **First Unique Character** - Find first non-repeating

Medium Level

- **Two Sum Variations** - Finding pairs with target sum
- **Group Anagrams** - Using sorted string as key
- **Subarray Sum Equals K** - Prefix sum + hash map
- **Longest Consecutive Sequence** - Union-find alternative
- **Top K Elements** - Frequency sorting
- **Custom Hash Function** - Design for specific problems
- **Rolling Hash** - String matching (Rabin-Karp)

Hard Level

- **Sliding Window + Hash Map** - Complex window with frequency tracking
- **Design Data Structures** - LRU Cache, LFU Cache
- **Minimum Window Substring** - Substring matching with conditions
- **All O(1) Data Structure** - Constant time operations

---

3. Linked List Patterns

Easy Level

- **Linear Traversal** - Simple iteration
- **Two Pointers (Fast-Slow)** - Detect cycle, find middle
- **Reversal** - Reverse entire list
- **Dummy Node Technique** - Simplify edge cases

Medium Level

- **Reversal in Groups** - Reverse k-node groups, rotation

- **Reversal in Groups** - Reverse k nodes at a time
- **Merge Two Lists** - Merge sorted lists
- **Reorder List** - Split, reverse, merge pattern
- **Remove Nth Node** - Two pointer with gap
- **Flatten Multi-level List** - DFS on linked list
- **Partition List** - Split based on value

Hard Level

- **Reverse Nodes in K-Group** - Complex reversal with conditions
- **Merge K Sorted Lists** - Using heap/priority queue
- **Copy List with Random Pointer** - Deep copy with hash map
- **LRU Cache Implementation** - Doubly linked list + hash map
- **Skip List** - Probabilistic data structure

---

4. Stack & Queue Patterns

Easy Level

- **Basic Stack Operations** - Push, pop, peek
- **Valid Parentheses** - Matching brackets
- **Monotonic Stack (Basic)** - Next greater element
- **Queue using Stacks** - Implement queue with two stacks
- **Stack using Queues** - Implement stack with queues

Medium Level

- **Min/Max Stack** - Stack with O(1) min/max query
- **Evaluate Expression** - Postfix/prefix evaluation
- **Decode String** - Nested pattern decoding
- **Daily Temperatures** - Monotonic stack application
- **Asteroid Collision** - Stack simulation
- **Remove K Digits** - Monotonic stack for minimum
- **132 Pattern** - Stack with max tracking

Hard Level

Hard Level

- **Largest Rectangle in Histogram** - Stack with area calculation
- **Trapping Rain Water** - Using stack approach
- **Basic Calculator** - Handle operators and parentheses
- **Max Stack** - Stack with O(log n) max operations
- **Sliding Window Maximum** - Deque pattern

---

5. Tree Patterns

Easy Level

- **Tree Traversal (DFS)** - Inorder, preorder, postorder
- **Tree Traversal (BFS)** - Level order
- **Height/Depth Calculation** - Recursive measurement
- **Path Sum** - Check if path equals target
- **Symmetric Tree** - Mirror checking
- **Invert Binary Tree** - Swap left and right
- **Merge Two Trees** - Combine trees

Medium Level

- **Bottom-Up DP on Trees** - Calculate from leaves to root
- **Top-Down DFS** - Pass information from parent to child
- **Lowest Common Ancestor (LCA)** - Find common ancestor
- **Serialize/Deserialize** - Convert tree to/from string
- **Vertical Order Traversal** - Grouping by columns
- **Right Side View** - BFS variation
- **Zigzag Level Order** - BFS with alternating direction
- **Construct Tree from Traversals** - Inorder + Preorder/Postorder
- **Diameter of Tree** - Longest path
- **Boundary Traversal** - Traverse tree boundary
- **Flatten Tree to Linked List** - Tree transformation
- **Sum Root to Leaf Numbers** - Path calculation

Hard Level

- **Path Sum III** - Any path in tree
- **Binary Tree Maximum Path Sum** - Consider all paths
- **Morris Traversal** - O(1) space traversal
- **Recover BST** - Fix swapped nodes
- **Binary Tree Cameras** - Greedy on trees
- **All Nodes Distance K** - BFS with parent tracking
- **Vertical Order with Multiple Constraints** - Complex sorting

---

6. Binary Search Tree (BST) Patterns

Easy Level

- **BST Search** - Utilize BST property
- **Validate BST** - Check BST property
- **Range Sum BST** - Sum in range

Medium Level

- **Insert into BST** - Maintain BST property
- **Delete from BST** - Handle three cases
- **Kth Smallest in BST** - Inorder traversal
- **Convert Sorted Array to BST** - Balanced BST construction
- **Trim BST** - Remove outside range nodes
- **Two Sum BST** - Use BST property

Hard Level

- **Largest BST Subtree** - Find largest valid BST
- **Recover BST** - Two nodes swapped
- **Serialize/Deserialize BST** - Optimized for BST
- **Count Smaller After Self** - BST with count

---

7. Binary Search Patterns

Easy Level

- **Basic Binary Search** - Search in sorted array
- **First/Last Occurrence** - Find boundaries
- **Search Insert Position** - Find where to insert
- **Perfect Square** - Binary search application

Medium Level

- **Search in Rotated Array** - Modified binary search
- **Find Peak Element** - Search in unsorted
- **Search in 2D Matrix** - Treat as 1D array
- **Sqrt(x)** - Binary search on answer
- **Find Minimum in Rotated Array** - Identify rotation point
- **Single Element in Sorted Array** - XOR property with binary search
- **Capacity to Ship Packages** - Binary search on answer
- **Koko Eating Bananas** - Binary search with validation

Hard Level

- **Median of Two Sorted Arrays** - Binary search on partition
- **Split Array Largest Sum** - Binary search on answer space
- **Aggressive Cows** - Binary search with greedy check
- **Kth Smallest in Multiplication Table** - 2D binary search
- **Find K-th Smallest Pair Distance** - Binary search + counting
- **Minimize Max Distance to Gas Station** - Binary search on doubles

---

8. Recursion & Backtracking Patterns

Easy Level

- **Simple Recursion** - Factorial, fibonacci
- **Generate Parentheses** - Build valid combinations
- **Subsets** - Generate all subsets
- **Power Set** - All possible combinations

Medium Level

- **Permutations** - All arrangements
- **Combinations** - Choose k from n
- **Combination Sum** - Target sum with candidates
- **Letter Combinations** - Phone number mapping
- **Word Search** - 2D grid backtracking
- **Palindrome Partitioning** - Split into palindromes
- **Restore IP Addresses** - Valid IP generation
- **Beautiful Arrangements** - Constrained permutations
- **Gray Code** - Generate gray code sequence
- **Flip Game II** - Game theory with backtracking

Hard Level

- **N-Queens** - Place queens on board
- **Sudoku Solver** - Fill valid sudoku
- **Expression Add Operators** - Insert operators for target
- **Remove Invalid Parentheses** - Minimum removals
- **Word Search II** - Multiple words in grid (with Trie)
- **Wildcard Matching** - Pattern matching
- **Regular Expression Matching** - With . and *
- **Matchsticks to Square** - Partition problem

---

9. Dynamic Programming Patterns

Easy Level

- **1D DP** - Fibonacci, climbing stairs
- **House Robber** - Non-adjacent selection
- **Maximum Subarray** - Kadane's algorithm
- **Min Cost Climbing Stairs** - Minimum cost path
- **Divisor Game** - Simple game theory

Medium Level

- **2D Grid DP** - Unique paths, minimum path sum
- **Knapsack (0/1)** - Include/exclude decisions
- **Unbounded Knapsack** - Unlimited items
- **Longest Common Subsequence (LCS)** - String matching
- **Longest Increasing Subsequence (LIS)** - Sequence problems
- **Coin Change** - Minimum coins for amount
- **Word Break** - Split string using dictionary
- **Decode Ways** - Count decoding methods
- **Unique Binary Search Trees** - Catalan number
- **Maximum Product Subarray** - Track min and max
- **Perfect Squares** - Minimum squares for sum
- **Target Sum** - Assign +/- to reach target
- **Partition Equal Subset Sum** - Subset sum variation
- **Last Stone Weight II** - Minimize difference
- **Ones and Zeroes** - 2D knapsack
- **Best Time to Buy/Sell Stock (All Variants)** - State machine DP

Hard Level

- **Edit Distance** - Transform one string to another
- **Palindrome Partitioning II** - Minimum cuts
- **Regular Expression Matching** - Pattern matching with * and .
- **Wildcard Matching** - Pattern with * and ?
- **Interleaving String** - Check if s3 is interleaving of s1, s2
- **Distinct Subsequences** - Count distinct subsequences
- **Burst Balloons** - Interval DP
- **Stone Game variations** - Game theory DP
- **Scramble String** - Check if scrambled
- **Maximum Profit in Job Scheduling** - Weighted interval scheduling
- **Minimum Difficulty of Job Schedule** - DP with monotonic stack
- **Cherry Pickup** - 3D DP on grid

- **Minimum Cost to Merge Stones** - Interval DP
- **Longest Valid Parentheses** - Parentheses DP
- **Count Different Palindromic Subsequences** - Complex counting
- **Freedom Trail** - Circular DP
- **Russian Doll Envelopes** - 2D LIS
- **Box Stacking** - 3D LIS variant

---

10. Graph Patterns

Easy Level

- **DFS Traversal** - Visit all nodes using recursion
- **BFS Traversal** - Level-wise traversal using queue
- **Connected Components** - Count islands/components
- **Cycle Detection (Undirected)** - Using DFS/BFS
- **Has Path** - Check if path exists

Medium Level

- **Cycle Detection (Directed)** - Using colors/states
- **Topological Sort** - Kahn's algorithm or DFS
- **Bipartite Check** - Two-coloring using BFS/DFS
- **Clone Graph** - Deep copy with hash map
- **Number of Islands** - DFS/BFS on grid
- **Surrounded Regions** - Boundary DFS/BFS
- **Course Schedule** - Topological sort application
- **Word Ladder** - BFS shortest path
- **Shortest Path in Binary Matrix** - BFS on grid
- **Rotting Oranges** - Multi-source BFS
- **Pacific Atlantic Water Flow** - Reverse thinking
- **Evaluate Division** - Graph with weights
- **Network Delay Time** - Dijkstra application

- **Redundant Connection** - Union-find for cycle
- **Accounts Merge** - Union-find for grouping

Hard Level

- **Dijkstra's Algorithm** - Shortest path (weighted)
- **Bellman-Ford** - Shortest path with negative weights
- **Floyd-Warshall** - All pairs shortest path
- **Prim's/Kruskal's** - Minimum spanning tree
- **Union-Find (Disjoint Set)** - Dynamic connectivity
- **Strongly Connected Components** - Kosaraju's/Tarjan's
- **Articulation Points & Bridges** - Critical connections
- **Traveling Salesman** - NP-hard optimization
- **Eulerian Path/Circuit** - Visit all edges once
- **Hamiltonian Path** - Visit all vertices once
- _A Algorithm_* - Heuristic search
- **Johnson's Algorithm** - All pairs with negative weights
- **Min Cut/Max Flow** - Ford-Fulkerson, Edmonds-Karp
- **Bipartite Matching** - Hungarian algorithm
- **Shortest Path with Obstacles** - Modified Dijkstra
- **Bus Routes** - Complex graph transformation
- **Word Ladder II** - All shortest paths
- **Alien Dictionary** - Topological sort with constraints

---

11. Greedy Patterns

Easy Level

- **Activity Selection** - Choose non-overlapping activities
- **Assign Cookies** - Match with constraints
- **Lemonade Change** - Cash change problem
- **Maximum Units on Truck** - Fractional knapsack variant

Medium Level

Medium Level

- **Jump Game** - Can reach end
- **Jump Game II** - Minimum jumps
- **Gas Station** - Circular array problem
- **Meeting Rooms II** - Minimum rooms needed
- **Non-overlapping Intervals** - Minimum removals
- **Partition Labels** - Split string into parts
- **Reorganize String** - Greedy arrangement
- **Task Scheduler** - CPU scheduling with cooldown
- **Remove Duplicate Letters** - Lexicographically smallest
- **Wiggle Subsequence** - Alternating sequence

Hard Level

- **Merge Intervals** - Complex merging logic
- **Insert Interval** - Insert and merge
- **Minimum Number of Arrows** - Burst balloons
- **Candy** - Distribution with constraints
- **Create Maximum Number** - Merge arrays optimally
- **Queue Reconstruction by Height** - Greedy with sorting
- **Split Array into Consecutive Subsequences** - Greedy grouping
- **IPO** - Maximum capital with constraints

---

12. Heap/Priority Queue Patterns

Easy Level

- **Kth Largest Element** - Using max heap
- **Last Stone Weight** - Simulation with heap
- **Relative Ranks** - Priority queue for ranking

Medium Level

- **Top K Frequent Elements** - Frequency + heap
- **Kth Largest in Stream** - Min heap of size k

- **Meeting Rooms II** - Min heap for end times
- **Find Median from Data Stream** - Two heaps
- **Reorganize String** - Heap for greedy selection
- **K Closest Points** - Distance heap
- **Kth Smallest Element in Sorted Matrix** - Min heap
- **Ugly Number II** - Multiple pointers with heap
- **Super Ugly Number** - Heap with multiple primes

Hard Level

- **Merge K Sorted Lists** - Min heap with list nodes
- **Sliding Window Median** - Two heaps + removal
- **IPO (Maximum Capital)** - Two heaps for projects
- **The Skyline Problem** - Complex heap operations
- **Find Median After K Operations** - Dynamic median
- **Trapping Rain Water II** - 3D water trap with heap
- **Swim in Rising Water** - Binary search or heap

---

13. Bit Manipulation Patterns

Easy Level

- **Check Bit Operations** - Get, set, clear bits
- **Count Set Bits** - Brian Kernighan's algorithm
- **Power of Two** - n & (n-1) == 0
- **Power of Four** - Additional check
- **Hamming Distance** - XOR + count bits
- **Reverse Bits** - Bit reversal

Medium Level

- **Single Number** - XOR properties
- **Single Number II** - Bits counting
- **Single Number III** - Two unique numbers

- **Subsets using Bits** - Iterate through all subsets
- **Missing Number** - XOR trick
- **Gray Code** - Generate sequence
- **Total Hamming Distance** - Count differences
- **Maximum XOR** - Greedy bit by bit
- **Bitwise AND of Range** - Common prefix
- **UTF-8 Validation** - Bit pattern checking

Hard Level

- **Maximum XOR of Two Numbers** - Trie on bits
- **Maximum XOR with Element from Array** - Offline queries
- **Count Triplets (XOR)** - Complex XOR relations
- **Minimum XOR Sum** - Assignment with XOR
- **Find XOR Sum of All Pairs** - Bitwise operations

---

14. Trie (Prefix Tree) Patterns

Easy Level

- **Implement Trie** - Insert, search, startsWith
- **Longest Common Prefix** - Using trie

Medium Level

- **Word Search II** - Backtracking + Trie
- **Design Add and Search Words** - With wildcard
- **Replace Words** - Dictionary with shortest root
- **Map Sum Pairs** - Trie with values
- **Implement Magic Dictionary** - Trie with one edit
- **Top K Frequent Words** - Trie + heap

Hard Level

- **Word Squares** - Build valid word squares
- **Palindrome Pairs** - Combine words to form palindromes

- **Stream of Characters** - Reverse trie matching
- **Concatenated Words** - Multiple words forming one
- **Maximum XOR** - Binary trie for numbers
- **Design Search Autocomplete** - Trie with frequency

---

15. Interval Patterns

Easy Level

- **Merge Intervals** - Combine overlapping intervals
- **Insert Interval** - Add and merge

Medium Level

- **Non-overlapping Intervals** - Minimum removals
- **Meeting Rooms** - Check availability
- **Meeting Rooms II** - Minimum rooms needed
- **Minimum Number of Arrows** - Burst balloons
- **Interval List Intersections** - Find overlaps
- **Remove Covered Intervals** - Remove subsets

Hard Level

- **Employee Free Time** - Merge across multiple schedules
- **Range Module** - Track and query ranges
- **My Calendar III** - K-booking problem
- **Count Days Between Dates** - Interval calculation
- **Data Stream as Disjoint Intervals** - Dynamic intervals

---

16. Matrix Patterns

Easy Level

- **Traverse Matrix** - Row/column iteration
- **Spiral Order** - Print in spiral
- **Transpose Matrix** - Swap rows and columns

- **Reshape Matrix** - Change dimensions

Medium Level

- **Rotate Image** - 90-degree rotation
- **Set Matrix Zeroes** - In-place modification
- **Search 2D Matrix** - Binary search
- **Search 2D Matrix II** - Staircase search
- **Word Search** - DFS on grid
- **Valid Sudoku** - Check constraints
- **Diagonal Traverse** - Zigzag diagonal
- **Kth Smallest in Sorted Matrix** - Binary search or heap

Hard Level

- **Sudoku Solver** - Constraint satisfaction
- **N-Queens** - Backtracking on board
- **Largest Rectangle** - Using monotonic stack
- **Maximal Rectangle** - Histogram approach
- **Maximal Square** - DP on matrix
- **Count Square Submatrices** - DP counting
- **Dungeon Game** - Reverse DP

---

17. String Matching Patterns

Easy Level

- **Naive String Matching** - Brute force
- **Two Pointer on Strings** - Palindrome check

Medium Level

- **KMP Algorithm** - Linear pattern matching
- **Rabin-Karp** - Rolling hash
- **Z-Algorithm** - Linear time pattern matching
- **Manacher's Algorithm** - Longest palindrome substring

- **Longest Palindromic Substring** - Expand around center

Hard Level

- **Aho-Corasick** - Multiple pattern matching
- **Suffix Array** - Pattern matching in text
- **Suffix Tree** - Complex string queries
- **Boyer-Moore** - Efficient pattern matching

---

18. Segment Tree Patterns

Medium Level

- **Range Sum Query (Mutable)** - Update and query
- **Range Minimum/Maximum Query** - Min/max in range
- **Count of Range Sum** - Count elements in range

Hard Level

- **Range Update Range Query** - Lazy propagation
- **2D Range Sum** - 2D segment tree
- **Persistent Segment Tree** - Historical queries
- **Dynamic Segment Tree** - Coordinate compression

---

19. Binary Indexed Tree (Fenwick Tree) Patterns

Medium Level

- **Range Sum Query (Mutable)** - Efficient updates
- **Count Smaller After Self** - Inversion counting
- **Reverse Pairs** - Counting inversions

Hard Level

- **2D Binary Indexed Tree** - 2D queries
- **Range Update Point Query** - Difference array
- **Count of Range Sum** - With BIT

---

20. Union-Find (Disjoint Set) Patterns

Medium Level

- **Number of Connected Components** - Count components
- **Redundant Connection** - Find extra edge
- **Accounts Merge** - Merge by email
- **Most Stones Removed** - Component counting
- **Satisfiability of Equality Equations** - Track equalities

Hard Level

- **Number of Islands II** - Dynamic island addition
- **Smallest String with Swaps** - Union-find + sorting
- **Minimize Malware Spread** - Critical node finding
- **Regions Cut by Slashes** - Grid union-find
- **Bricks Falling When Hit** - Reverse union-find

---

21. Monotonic Queue/Deque Patterns

Medium Level

- **Sliding Window Maximum** - Max in each window
- **Longest Continuous Subarray** - Deque for min/max
- **Shortest Subarray with Sum >= K** - Prefix sum + deque

Hard Level

- **Max Value of Equation** - Complex deque usage
- **Constrained Subsequence Sum** - DP with deque

---

22. Reservoir Sampling

Medium Level

- **Random Pick Index** - Uniform random selection
- **Linked List Random Node** - Unknown size sampling
- **Random Pick with Weight** - Weighted sampling

- **Random Pick with Weight** - Weighted sampling

Hard Level

- **Random Pick with Blacklist** - Exclude elements
- **K Random Elements from Stream** - Maintain K elements

---

23. Math & Number Theory Patterns

Easy Level

- **Prime Number Check** - Basic primality
- **GCD/LCM** - Euclidean algorithm
- **Factorial** - Simple calculation
- **Palindrome Number** - Reverse and compare

Medium Level

- **Sieve of Eratosthenes** - Generate primes
- **Prime Factorization** - Factor a number
- **Modular Arithmetic** - Modulo operations
- **Fast Exponentiation** - Power in log time
- **Chinese Remainder Theorem** - System of congruences
- **Catalan Numbers** - Combinatorial problems
- **Pascal's Triangle** - Binomial coefficients
- **Ugly Number** - Numbers with specific factors

Hard Level

- **Miller-Rabin** - Probabilistic primality
- **Pollard's Rho** - Integer factorization
- **Extended Euclidean** - Multiplicative inverse
- **Fermat's Little Theorem** - Modular applications
- **Combinatorics** - Advanced counting

---

24. Geometry Patterns

Medium Level

Medium Level

- **Convex Hull** - Graham scan, Jarvis march
- **Line Intersection** - Check if lines cross
- **Point in Polygon** - Ray casting
- **Closest Pair of Points** - Divide and conquer
- **Rectangle Overlap** - Check intersection

Hard Level

- **Voronoi Diagram** - Nearest neighbor regions
- **Sweep Line Algorithm** - Event-driven geometry
- **Rotating Calipers** - Diameter of convex hull

---

25. Game Theory Patterns

Easy Level

- **Nim Game** - Basic game theory
- **Stone Game** - Optimal play

Medium Level

- **Divisor Game** - DP game theory
- **Cat and Mouse** - Graph game
- **Stone Game II/III/IV** - Variations

Hard Level

- **Minimax Algorithm** - Optimal decision
- **Alpha-Beta Pruning** - Optimized minimax
- **Sprague-Grundy Theorem** - Impartial games
- **Game on Tree** - Tree game theory

---

26. Randomized Algorithms

Medium Level

- **Shuffle Array** - Fisher-Yates shuffle

- **Random Pick with Weight** - Prefix sum + binary search
- **Random Point in Rectangle** - Uniform distribution

Hard Level

- **Random Point in Non-overlapping Rectangles** - Complex sampling
- **Monte Carlo Methods** - Probabilistic algorithms
- **Las Vegas Algorithms** - Randomized correct algorithms

---

27. Divide and Conquer Patterns

Medium Level

- **Merge Sort** - Sorting by division
- **Quick Sort** - Partition-based sorting
- **Binary Search** - Divide search space
- **Maximum Subarray** - Divide and conquer approach
- **Closest Pair of Points** - Geometric divide

Hard Level

- **Median of Medians** - Linear time selection
- **Strassen's Matrix Multiplication** - Fast matrix multiply
- **Karatsuba Algorithm** - Fast multiplication

---

28. String Processing Patterns

Easy Level

- **Reverse String** - Two pointers
- **Valid Palindrome** - Filter + check
- **Anagram Check** - Frequency comparison

Medium Level

- **String Compression** - Run-length encoding
- **Group Shifted Strings** - Normalization
- **Longest Substring Without Repeating** - Sliding window

- **Longest Substring Without Repeating** - Sliding window
- **Minimum Window Substring** - Two pointers + map
- **Longest Repeating Character Replacement** - Sliding window
- **Valid Anagram** - Multiple approaches

Hard Level

- **Shortest Palindrome** - KMP variation
- **Text Justification** - Greedy formatting
- **Integer to English Words** - Complex string building
- **Encode and Decode Strings** - Delimiter design

---

29. Coordinate Compression

Medium Level

- **Rank Transform** - Map values to ranks
- **My Calendar I/II/III** - Compress time points

Hard Level

- **Rectangle Area II** - Sweep line with compression
- **The Skyline Problem** - Coordinate compression

---

30. Meet in the Middle

Hard Level

- **Partition to K Equal Sum Subsets** - Split search space
- **Closest Subsequence Sum** - Two halves approach
- **4Sum II** - Split into pairs

---

31. Rolling Hash

Medium Level

- **Repeated DNA Sequences** - Hash sliding window
- **Rabin-Karp String Matching** - Pattern matching

Hard Level

- **Longest Duplicate Substring** - Binary search + rolling hash
- **Distinct Echo Substrings** - Hash-based detection

---

32. Sparse Table

Hard Level

- **Range Minimum Query (Static)** - O(1) query after preprocessing
- **Range GCD Query** - Sparse table application

---

33. Heavy-Light Decomposition

Hard Level

- **Path Queries on Tree** - Tree path queries
- **Tree Query with Updates** - Efficient tree operations

---

34. Centroid Decomposition

Hard Level

- **Count Pairs with Distance K** - Tree decomposition
- **Path Queries in Trees** - Logarithmic depth decomposition

---

35. Square Root Decomposition

Medium Level

- **Range Sum Query** - Block-based queries
- **Mo's Algorithm** - Query ordering technique

---

36. Multi-threading Patterns

Medium Level

- **Print in Order** - Synchronization

- **Print FooBar Alternately** - Thread coordination
- **Producer-Consumer** - Classic concurrency

Hard Level

- **Dining Philosophers** - Deadlock prevention
- **Traffic Light Controlled Intersection** - Complex synchronization

---

37. Simulation Patterns

Easy Level

- **Robot Return to Origin** - Track position
- **Walking Robot Simulation** - Grid simulation

Medium Level

- **Spiral Matrix** - Generate spiral
- **Robot Bounded in Circle** - Cycle detection
- **Snakes and Ladders** - BFS simulation

Hard Level

- **Race Car** - BFS with states
- **Cherry Pickup** - Grid simulation with DP

---

38. State Machine Patterns

Medium Level

- **Best Time to Buy/Sell Stock** - State transitions
- **UTF-8 Validation** - State-based validation
- **Design Log Storage** - State management

Hard Level

- **Regular Expression Matching** - State machine DP
- **Valid Number** - Complex state machine

---

39. Two Heaps Patt

39. Two Heaps Pattern

Hard Level

- **Find Median from Data Stream** - Max heap + min heap
- **Sliding Window Median** - Two heaps with removal
- **IPO** - Capital and profit heaps

---

40. Prefix/Suffix Sum Patterns

Easy Level

- **Running Sum** - Accumulate sums
- **Pivot Index** - Left sum = right sum

Medium Level

- **Subarray Sum Equals K** - Prefix sum + hash map
- **Continuous Subarray Sum** - Prefix mod
- **Product of Array Except Self** - Prefix/suffix product

Hard Level

- **Maximum Sum of 3 Non-Overlapping Subarrays** - Multiple prefix arrays
- **Count of Range Sum** - Prefix + merge sort

Learning Strategy

Phase 1: Foundation (Weeks 1-4)

- Array & String basics
- Two Pointers
- Sliding Window
- Hash Map
- Basic Recursion

Phase 2: Core Data Structures (Weeks 5-8)

- Linked List
- Stack & Queue

- Binary Search
- Trees (BFS/DFS)
- Basic Graphs

Phase 3: Advanced Patterns (Weeks 9-12)

- Dynamic Programming
- Backtracking
- Greedy Algorithms
- Heaps
- Advanced Graphs

Phase 4: Specialized (Weeks 13-16)

- Trie
- Union-Find
- Segment Tree
- Bit Manipulation
- String Matching

Phase 5: Expert (Weeks 17+)

- Advanced Trees (Heavy-Light, Centroid)
- Game Theory
- Math & Number Theory
- Geometry
- Randomized Algorithms

Practice Approach:

1. **Learn pattern concept** - Understand when and why to use it
2. **Solve 1 easy** - Build confidence
3. **Solve 3 medium** - Recognize variations
4. **Attempt 1 hard** - Push limits
5. **Revisit weekly** - Spaced repetition
6. **Create templates** - Speed up implementation

Pro Tips:

- Focus on **understanding patterns** not memorizing solutions
- **One pattern per week** - Deep mastery over breadth
- **Code templates** - Create reusable pattern templates
- **Pattern journal** - Document when each pattern applies
- **Daily practice** - 30 minutes minimum, consistency matters
- **Review cycle** - Day 1, Day 3, Day 7, Day 30
- **Teach others** - Best way to solidify understanding
- **Mock interviews** - Practice under time pressure

---

41. Suffix Automaton

Hard Level

- **Count Distinct Substrings** - Linear time counting
- **Longest Common Substring** - Multiple strings
- **String Matching Queries** - Efficient pattern matching

---

42. Palindrome Patterns (Specialized)

Easy Level

- **Valid Palindrome** - Two pointers check
- **Palindrome Number** - Reverse comparison

Medium Level

- **Longest Palindromic Substring** - Expand around center
- **Palindrome Partitioning** - Backtracking
- **Palindrome Linked List** - Reverse and compare
- **Valid Palindrome II** - Allow one deletion
- **Shortest Palindrome** - Add characters to front

Hard Level

- **Palindrome Pairs** - Hash map or trie

- **Palindrome Partitioning II** - DP with minimum cuts
- **Super Palindromes** - Number theory + palindrome
- **Count Different Palindromic Subsequences** - DP counting
- **Longest Chunked Palindrome** - Greedy decomposition

---

43. Parentheses Patterns

Easy Level

- **Valid Parentheses** - Stack matching
- **Generate Parentheses** - Backtracking

Medium Level

- **Longest Valid Parentheses** - DP or stack
- **Minimum Add to Make Valid** - Greedy counting
- **Score of Parentheses** - Stack with scoring

Hard Level

- **Remove Invalid Parentheses** - BFS or backtracking
- **Different Ways to Add Parentheses** - Divide and conquer
- **Check Valid Parenthesis String** - Greedy with range

---

44. Subarray/Substring Patterns

Easy Level

- **Maximum Subarray** - Kadane's algorithm
- **Subarray Sum Equals K** - Prefix sum + hash

Medium Level

- **Longest Substring Without Repeating** - Sliding window
- **Minimum Size Subarray Sum** - Two pointers
- **Maximum Length of Repeated Subarray** - DP or rolling hash
- **Subarray Product Less Than K** - Sliding window
- **Number of Subarrays with Bounded Max** - Contribution technique

- **Count Subarrays with Score Less Than K** - Sliding window

Hard Level

- **Subarrays with K Different Integers** - Sliding window (at most K trick)
- **Shortest Subarray with Sum at Least K** - Deque optimization
- **Count Subarrays with Fixed Bounds** - Complex sliding window
- **Maximum Sum of 3 Non-Overlapping Subarrays** - DP with tracking

---

45. Subsequence Patterns

Medium Level

- **Longest Increasing Subsequence** - DP or binary search
- **Is Subsequence** - Two pointers
- **Number of Matching Subsequences** - Hash map with pointers
- **Longest Common Subsequence** - 2D DP

Hard Level

- **Distinct Subsequences** - DP counting
- **Shortest Common Supersequence** - LCS variation
- **Count Different Palindromic Subsequences** - Complex DP
- **Number of Longest Increasing Subsequence** - DP with count

---

46. Cycle Detection Patterns

Easy Level

- **Linked List Cycle** - Fast and slow pointers
- **Happy Number** - Cycle detection

Medium Level

- **Find Duplicate Number** - Floyd's cycle detection on array
- **Circular Array Loop** - Cycle in array
- **Detect Cycle in Directed Graph** - DFS with colors

Hard Level

- **Find All Duplicates** - In-place marking
- **Course Schedule II** - Topological sort with cycle check

---

47. Sorting Patterns

Easy Level

- **Bubble Sort** - Simple comparison sort
- **Selection Sort** - Find min and swap
- **Insertion Sort** - Build sorted array

Medium Level

- **Merge Sort** - Divide and conquer O(n log n)
- **Quick Sort** - Partition-based sorting
- **Heap Sort** - Using heap structure
- **Counting Sort** - Integer sorting O(n+k)
- **Bucket Sort** - Distribution sort
- **Radix Sort** - Digit-by-digit sort
- **Custom Sort** - Comparator-based sorting
- **Sort Colors** - Dutch national flag

Hard Level

- **Pancake Sorting** - Limited operations
- **Sort List** - Merge sort on linked list
- **Wiggle Sort II** - Complex in-place arrangement

---

48. Matrix Chain Multiplication Pattern

Hard Level

- **Matrix Chain Multiplication** - Optimal parenthesization
- **Burst Balloons** - Similar interval DP
- **Minimum Cost to Merge Stones** - K-way merge

---

49. Probability & Expected Value

Medium Level

- **Soup Servings** - Probability DP
- **New 21 Game** - Sliding window probability
- **Knight Probability** - DP on board

Hard Level

- **Airplane Seat Assignment** - Mathematical probability
- **Random Pick with Blacklist** - Uniform distribution

---

50. Constructive Algorithms

Medium Level

- **Reconstruct Itinerary** - Eulerian path
- **Construct Target Array** - Reverse engineering
- **Find Original Array** - Reverse doubling

Hard Level

- **Construct K Palindrome Strings** - Greedy construction
- **Minimum Number of K Consecutive Bit Flips** - Greedy with markers

---

51. Bitmask DP

Medium Level

- **Partition to K Equal Sum Subsets** - Bitmask state
- **Fair Distribution of Cookies** - Bitmask backtracking

Hard Level

- **Traveling Salesman** - Bitmask DP on cities
- **Maximum Students Taking Exam** - Bitmask for rows
- **Number of Ways to Wear Different Hats** - Bitmask on items
- **Minimum Cost to Connect Two Groups** - Bitmask matching

52. Digit DP

Hard Level

- **Count Numbers with Unique Digits** - Combinatorics or DP
- **Numbers At Most N Given Digit Set** - Build numbers digit by digit
- **Numbers with Repeated Digits** - Digit DP with state
- **Count Special Integers** - No repeated digits

---

53. Line Sweep Algorithm

Medium Level

- **Meeting Rooms II** - Sweep with events
- **My Calendar I/II/III** - Interval booking

Hard Level

- **The Skyline Problem** - Complex sweep with heap
- **Rectangle Area II** - Sweep with coordinate compression
- **Perfect Rectangle** - Sweep with validation

---

54. Minimax with Alpha-Beta Pruning

Hard Level

- **Predict the Winner** - Game theory with minimax
- **Stone Game variations** - Optimal play
- **Cat and Mouse** - Graph game with states

---

55. 2-SAT Problem

Hard Level

- **Satisfiability of Equality Equations** - Graph-based 2-SAT
- **Escape a Large Maze** - Implicit graph with constraints

---

56. Offline Queries Pattern

Hard Level

- **Range Sum Query 2D - Mutable** - Segment tree or BIT
- **Count of Smaller Numbers After Self** - Merge sort or BIT
- **Reverse Pairs** - Modified merge sort

---

57. Frequency-Based Patterns

Easy Level

- **Majority Element** - Boyer-Moore voting
- **Most Common Word** - Hash map frequency

Medium Level

- **Top K Frequent Elements** - Heap or bucket sort
- **Sort Characters by Frequency** - Frequency sorting
- **Top K Frequent Words** - Heap with custom comparator

Hard Level

- **First Unique Character in Stream** - Queue + hash map
- **LFU Cache** - Complex frequency tracking

---

58. Interval Scheduling Patterns

Medium Level

- **Non-overlapping Intervals** - Greedy sorting
- **Minimum Number of Arrows** - Interval merging
- **Maximum Profit in Job Scheduling** - DP on intervals

Hard Level

- **Employee Free Time** - Merge all schedules
- **My Calendar III** - K-booking with sweep

---

59. Expression Evaluation Patterns

Medium Level

- **Basic Calculator** - Stack with operators
- **Evaluate Reverse Polish Notation** - Stack-based
- **Different Ways to Add Parentheses** - Divide and conquer

Hard Level

- **Basic Calculator III** - Handle parentheses and operators
- **Parse Lisp Expression** - Recursive parsing
- **Ternary Expression Parser** - Stack-based parsing

---

60. Partition Problems

Medium Level

- **Partition Equal Subset Sum** - Subset sum DP
- **Partition to K Equal Sum Subsets** - Backtracking with bitmask
- **Partition Array for Maximum Sum** - DP with windows

Hard Level

- **Split Array Largest Sum** - Binary search + greedy
- **Minimum Difficulty of Job Schedule** - DP with constraints
- **Last Stone Weight II** - Minimize difference

---

61. Stock Trading Patterns

Easy Level

- **Best Time to Buy and Sell Stock** - Single transaction
- **Best Time to Buy and Sell Stock II** - Unlimited transactions

Medium Level

- **Best Time to Buy and Sell Stock III** - Two transactions
- **Best Time to Buy and Sell Stock with Cooldown** - State machine DP

Hard Level

Hard Level

- **Best Time to Buy and Sell Stock IV** - K transactions
- **Best Time to Buy and Sell with Transaction Fee** - Fee consideration

---

62. LIS (Longest Increasing Subsequence) Variants

Medium Level

- **Longest Increasing Subsequence** - $O(n^2)$ DP or $O(n \log n)$
- **Number of LIS** - Count all longest
- **Longest Increasing Path in Matrix** - DFS with memoization

Hard Level

- **Russian Doll Envelopes** - 2D LIS
- **Maximum Height by Stacking Cuboids** - 3D LIS variant
- **Make Array Strictly Increasing** - LIS with replacements

---

63. Grid/Matrix DP Patterns

Easy Level

- **Unique Paths** - Simple grid DP
- **Minimum Path Sum** - Cost accumulation

Medium Level

- **Unique Paths II** - With obstacles
- **Minimum Path Sum** - Weighted paths
- **Maximal Square** - Find largest square
- **Dungeon Game** - Reverse DP

Hard Level

- **Cherry Pickup** - Two paths simultaneously
- **Cherry Pickup II** - Multiple robots
- **Minimum Cost Homecoming** - Grid with costs

---

64. Tree Construction Patterns

64. Tree Construction Patterns

Medium Level

- **Construct Binary Tree from Preorder and Inorder** - Recursive construction
- **Construct Binary Tree from Inorder and Postorder** - Similar approach
- **Construct BST from Preorder** - Use BST property
- **Convert Sorted Array to BST** - Balanced construction

Hard Level

- **Serialize and Deserialize Binary Tree** - String encoding
- **Serialize and Deserialize N-ary Tree** - Complex encoding
- **Recover BST** - Fix two swapped nodes

---

65. Path Finding in Grid

Medium Level

- **Number of Islands** - DFS/BFS counting
- **Max Area of Island** - DFS with size
- **Shortest Path in Binary Matrix** - BFS
- **As Far from Land** - Multi-source BFS

Hard Level

- **Shortest Path in Grid with Obstacles** - BFS with state
- **Minimum Obstacle Removal** - 0-1 BFS
- **Swim in Rising Water** - Binary search or Dijkstra
- **Trapping Rain Water II** - 3D version with heap

---

66. Parenthesis Matching Advanced

Hard Level

- **Valid Parenthesis String** - Greedy with range
- **Minimum Insertions to Balance** - Greedy counting
- **Minimum Remove to Make Valid** - Stack tracking

---

67. Median Finding Patterns

Medium Level

- **Find Median from Data Stream** - Two heaps
- **Sliding Window Median** - Two heaps with removal

Hard Level

- **Median of Two Sorted Arrays** - Binary search partition
- **Find Kth Smallest Pair Distance** - Binary search + counting

---

68. LCA (Lowest Common Ancestor) Variants

Easy Level

- **LCA of Binary Tree** - DFS approach
- **LCA of BST** - Use BST property

Medium Level

- **LCA of Deepest Leaves** - Track depth
- **All Nodes Distance K** - BFS with parent pointers

Hard Level

- **Binary Lifting** - O(log n) LCA queries
- **LCA in DAG** - Topological sort based

---

69. Jump Game Patterns

Medium Level

- **Jump Game** - Can reach end (greedy)
- **Jump Game II** - Minimum jumps (BFS/greedy)
- **Jump Game III** - Reach value 0 (BFS)
- **Jump Game IV** - Jump to same values (BFS)

Hard Level

- **Jump Game V** - Jump with restrictions (DP)

- **Jump Game VI** - Maximum score (DP + deque)
- **Jump Game VII** - Binary jumps (BFS)

---

70. Design Problems (System Design with DS)

Medium Level

- **Design HashSet** - Hash function + buckets
- **Design HashMap** - Key-value storage
- **Design Circular Queue** - Array-based queue
- **Design Browser History** - Stack or doubly linked list
- **Design Underground System** - Hash maps for tracking

Hard Level

- **LRU Cache** - Hash map + doubly linked list
- **LFU Cache** - Hash maps + doubly linked list with frequency
- **Design Twitter** - Multiple data structures
- **Design Search Autocomplete System** - Trie + heap
- **Design File System** - Trie or hash map
- **Design In-Memory File System** - Tree structure
- **Design Phone Directory** - Set with recycling
- **All O(1) Data Structure** - Multiple hash maps + DLL

Complete Pattern Count Summary

Total Unique Patterns: 300+

By Difficulty:

- **Easy Patterns**: ~60
- **Medium Patterns**: ~140
- **Hard Patterns**: ~100+

By Category:

- **Core Data Structures**: 80+ patterns

- **Core Data Structures**: 80+ patterns
- **Algorithm Techniques**: 90+ patterns
- **Advanced Topics**: 70+ patterns
- **Specialized**: 60+ patterns

Mastery Roadmap (12-Month Plan)

Month 1-2: Foundations

- Arrays, Strings, Two Pointers
- Hash Maps, Sets
- Basic Recursion
- **Goal**: Solve 50 easy problems

Month 3-4: Core Structures

- Linked Lists
- Stacks & Queues
- Binary Search
- Trees (BFS, DFS)
- **Goal**: 40 medium problems

Month 5-6: Intermediate Patterns

- Backtracking
- Basic DP (1D)
- Graphs (DFS, BFS)
- Heaps
- **Goal**: 50 medium problems

Month 7-8: Advanced DP

- 2D DP
- Tree DP
- Graph DP
- Knapsack variations
- **Goal**: 30 medium, 10 hard

Month 9-10: Advanced Structures

- Trie
- Union-Find
- Segment Tree
- String matching
- **Goal**: 20 medium, 15 hard

Month 11-12: Expert Level

- Advanced graph algorithms
- Game theory
- Bit manipulation
- Math & geometry
- **Goal**: 40 hard problems

Total Target: 300+ problems across all difficulties

Pattern Recognition Cheat Sheet

When you see...

- **"In sorted array"** → Binary Search
- **"All subsets/combinations"** → Backtracking
- **"Shortest path (unweighted)"** → BFS
- **"Shortest path (weighted)"** → Dijkstra
- **"Optimize/maximize/minimize"** → DP or Greedy
- **"K largest/smallest"** → Heap
- **"Find cycle"** → Fast-slow pointers or DFS
- **"Prefix"** → Trie
- **"Range query"** → Segment Tree or BIT
- **"Disjoint sets"** → Union-Find
- **"Sliding"** → Sliding Window or Deque
- **"Consecutive"** → Hash Map with prefix

- **"Palindrome"** → Expand center or DP
- **"Parentheses"** → Stack
- **"Tree traversal"** → DFS or BFS
- **"In-place with cycle"** → Cyclic sort
- **"Design data structure"** → Hash Map + specific structure

---

Practice Resources Priority

1. **LeetCode** - Primary platform (patterns tagged)
2. **Codeforces** - For competitive programming
3. **HackerRank** - Interview prep
4. **GeeksforGeeks** - Concept explanations
5. **InterviewBit** - Structured path

Final Tips for Backend Developers

Prioritize These Patterns:

1. **Hash Maps** - Most used in backend
2. **Trees & Graphs** - Database structures
3. **DP** - Optimization problems
4. **String Processing** - API data handling
5. **Heaps** - Task scheduling

Less Priority (Initially):

- Geometry
- Game Theory
- Advanced Math
- Bit Manipulation (except basics)

Connect to Backend Work:

- **Caching** = LRU/LFU patterns
- **Rate Limiting** = Sliding window

- **Database Queries** = Tree/Graph traversal
- **Load Balancing** = Greedy algorithms
- **Data Processing** = Array/String patterns

Remember: You don't need to master ALL patterns. Focus on **most frequent 50-60 patterns** for interviews and real work!