Striver's A2Z DSA Sheet - Complete 450+ Problems Organization

📊 Total Count: 450+ Problems

🎯 PATTERN-WISE ORGANIZATION WITH DIFFICULTY LEVELS

PATTERN 1: BASIC PROGRAMMING & MATHEMATICS

Total Problems: ~35-40

🟢 Easy (25-30 problems)

- **Basic I/O Operations**
  - Data types, variables, input/output
  - Pattern printing (stars, numbers, alphabets)
  - Basic arithmetic operations
- **Mathematical Fundamentals**
  - Check if number is prime
  - Find GCD/LCM of two numbers
  - Check if number is palindrome
  - Count digits in a number
  - Reverse a number
  - Check if number is Armstrong number
  - Print all divisors of a number

🟡 Medium (8-10 problems)

- **Advanced Mathematics**
  - Sieve of Eratosthenes
  - Power calculation (x^n)
  - Square root using binary search
  - Check if number is perfect square

🔴 Hard (2-5 problems)

- **Complex Mathematical Problems**
  - Fast exponentiation

- Modular arithmetic problems

PATTERN 2: SORTING ALGORITHMS

Total Problems: ~25-30

🟢 Easy (15-18 problems)

- **Basic Sorting Understanding**
  - Selection Sort implementation
  - Bubble Sort implementation
  - Insertion Sort implementation
  - Merge two sorted arrays
  - Sort array of 0s, 1s, and 2s

🟡 Medium (8-10 problems)

- **Advanced Sorting Concepts**
  - Merge Sort implementation
  - Quick Sort implementation
  - Count inversions in array
  - Sort array by frequency

🔴 Hard (2-4 problems)

- **Complex Sorting Problems**
  - External sorting concepts
  - Sorting with custom comparators

PATTERN 3: ARRAY MANIPULATION

Total Problems: ~60-70

🟢 Easy (30-35 problems)

- **Basic Array Operations**
  - Find largest/smallest element
  - Second largest element
  - Check if array is sorted

- Remove duplicates from sorted array
- Left/Right rotate array by k positions
- Find missing number in array
- Move zeros to end
- Linear search implementation
- Find union of two sorted arrays

🟡 Medium (25-30 problems)

- **Two Pointers Pattern**
  - Two Sum problem
  - Three Sum problem
  - Four Sum problem
  - Container with most water
  - Sort colors (Dutch National Flag)
  - Remove duplicates from sorted array II
- **Sliding Window Pattern**
  - Maximum sum subarray of size k
  - Longest subarray with sum k
  - Count subarrays with given sum
- **Prefix Sum Pattern**
  - Subarray sum equals k
  - Maximum subarray sum (Kadane's algorithm)
  - Product of array except self

🔴 Hard (8-12 problems)

- **Advanced Array Problems**
  - Trapping rain water
  - Minimum window substring
  - Sliding window maximum
  - Longest substring without repeating characters
  - Median of two sorted arrays

PATTERN 4: BINARY SEARCH

Total Problems: ~35-40

🟢 Easy (15-18 problems)

- **Basic Binary Search**
  - Binary search implementation
  - Search insert position
  - Find first and last position of element
  - Search in rotated sorted array I
  - Find peak element
  - Find minimum in rotated sorted array

🟡 Medium (15-18 problems)

- **Search in Answer Space**
  - Square root using binary search
  - Find nth root of number
  - Koko eating bananas
  - Minimum days to make m bouquets
  - Smallest divisor given threshold
  - Capacity to ship packages within d days

🔴 Hard (5-8 problems)

- **Advanced Binary Search**
  - Median of two sorted arrays
  - Kth element of two sorted arrays
  - Aggressive cows problem
  - Book allocation problem

PATTERN 5: STRING MANIPULATION

Total Problems: ~30-35

🟢 Easy (15-18 problems)

- **Basic String Operations**

- Reverse a string
- Check if string is palindrome
- Count vowels and consonants
- Remove characters from string
- Check if strings are anagrams
- Longest common prefix

🟡 **Medium (12-15 problems)**

- **String Algorithms**
  - KMP algorithm for pattern matching
  - Rabin-Karp algorithm
  - Z algorithm
  - Minimum window substring
  - Group anagrams

🔴 **Hard (3-5 problems)**

- **Advanced String Problems**
  - Edit distance
  - Distinct subsequences
  - Wildcard pattern matching

PATTERN 6: LINKED LIST OPERATIONS

Total Problems: ~30-35

🟢 **Easy (15-18 problems)**

- **Basic Linked List Operations**
  - Insert/Delete at beginning, middle, end
  - Search in linked list
  - Find length of linked list
  - Reverse a linked list
  - Find middle of linked list
  - Detect cycle in linked list

🟡 Medium (12-15 problems)

- **Two Pointers in Linked List**
  - Remove nth node from end
  - Intersection of two linked lists
  - Add two numbers represented as linked lists
  - Merge two sorted linked lists
  - Remove duplicates from sorted linked list

🔴 Hard (3-5 problems)

- **Advanced Linked List Problems**
  - Merge k sorted linked lists
  - Reverse nodes in k-group
  - Clone linked list with random pointers

PATTERN 7: RECURSION & BACKTRACKING

Total Problems: ~40-45

🟢 Easy (18-20 problems)

- **Basic Recursion**
  - Factorial using recursion
  - Fibonacci using recursion
  - Print numbers 1 to n
  - Sum of first n natural numbers
  - Print all subsequences of string
  - Tower of Hanoi

🟡 Medium (18-20 problems)

- **Backtracking Problems**
  - Generate all permutations
  - Generate all combinations
  - Combination sum problems
  - Subset sum problems

- Word search in grid
- Generate parentheses

🔴 **Hard (4-7 problems)**

- **Complex Backtracking**
  - N-Queens problem
  - Sudoku solver
  - Word break problem
  - Palindrome partitioning

PATTERN 8: BIT MANIPULATION

Total Problems: ~20-25

🟢 **Easy (12-15 problems)**

- **Basic Bit Operations**
  - Check if number is power of 2
  - Count set bits in number
  - Find the odd occurring element
  - Swap two numbers using XOR
  - Check if ith bit is set

🟡 **Medium (6-8 problems)**

- **Intermediate Bit Problems**
  - Find two odd occurring elements
  - Bit difference between two numbers
  - Maximum XOR of two numbers in array

🔴 **Hard (2-4 problems)**

- **Advanced Bit Manipulation**
  - Maximum XOR subarray
  - Minimum XOR of two elements

PATTERN 9: STACK & QUEUE

Total Problems: ~35-40

🟢 Easy (18-20 problems)

- **Basic Stack Operations**
  - Implement stack using arrays/linked lists
  - Valid parentheses
  - Implement queue using stacks
  - Implement stack using queues

🟡 Medium (15-18 problems)

- **Stack/Queue Applications**
  - Next greater element
  - Previous smaller element
  - Largest rectangle in histogram
  - Sliding window maximum
  - LRU Cache implementation

🔴 Hard (2-4 problems)

- **Advanced Stack/Queue Problems**
  - Maximum rectangle in binary matrix
  - Implement min stack

PATTERN 10: TREE ALGORITHMS

Total Problems: ~50-55

🟢 Easy (25-28 problems)

- **Basic Tree Operations**
  - Tree traversals (Inorder, Preorder, Postorder)
  - Level order traversal
  - Height/Depth of binary tree
  - Diameter of binary tree
  - Check if tree is balanced
  - Mirror of binary tree

🟡 Medium (20-22 problems)

- **Tree Properties & Algorithms**
  - Lowest common ancestor
  - Path sum problems
  - Construct tree from traversals
  - Serialize and deserialize binary tree
  - Vertical order traversal
  - Top/Bottom view of binary tree

🔴 Hard (5-8 problems)

- **Advanced Tree Problems**
  - Binary tree maximum path sum
  - Recover binary search tree
  - Count complete tree nodes
  - Binary tree cameras

PATTERN 11: GRAPH ALGORITHMS

Total Problems: ~40-45

🟢 Easy (15-18 problems)

- **Basic Graph Concepts**
  - BFS implementation
  - DFS implementation
  - Detect cycle in undirected graph
  - Number of connected components
  - Graph representation methods

🟡 Medium (20-22 problems)

- **Graph Traversal Applications**
  - Detect cycle in directed graph
  - Topological sorting
  - Shortest path in unweighted graph

- Clone graph
- Course schedule problems
- Number of islands

🔴 **Hard (5-8 problems)**

- **Advanced Graph Algorithms**
  - Dijkstra's shortest path
  - Bellman-Ford algorithm
  - Floyd-Warshall algorithm
  - Minimum spanning tree (Kruskal's, Prim's)
  - Strongly connected components

PATTERN 12: DYNAMIC PROGRAMMING

Total Problems: ~50-55

🟢 **Easy (20-22 problems)**

- **1D DP Problems**
  - Climbing stairs
  - Fibonacci with memoization
  - House robber
  - Maximum sum with no adjacent elements
  - Coin change (count ways)

🟡 **Medium (25-28 problems)**

- **2D DP Problems**
  - Unique paths in grid
  - Minimum path sum
  - Longest common subsequence
  - Edit distance
  - Knapsack problems
  - Palindromic subsequences

🔴 **Hard (5-8 problems)**

- **Advanced DP Problems**
  - Longest increasing subsequence
  - Matrix chain multiplication
  - Egg dropping problem
  - Burst balloons

PATTERN 13: GREEDY ALGORITHMS

Total Problems: ~25-30

🟢 Easy (12-15 problems)

- **Basic Greedy Problems**
  - Activity selection problem
  - Fractional knapsack
  - Minimum coins needed
  - Find minimum platforms

🟡 Medium (10-12 problems)

- **Intermediate Greedy**
  - Job scheduling with deadlines
  - Minimum number of meetings
  - Huffman coding
  - Gas station problem

🔴 Hard (3-5 problems)

- **Advanced Greedy Problems**
  - Candy distribution
  - Jump game variations

PATTERN 14: HEAP/PRIORITY QUEUE

Total Problems: ~20-25

🟢 Easy (10-12 problems)

- **Basic Heap Operations**

- Implement min/max heap
- Kth largest/smallest element
- Sort array using heap

🟡 Medium (8-10 problems)

- **Heap Applications**
  - Top K frequent elements
  - Merge K sorted arrays
  - Find median from data stream

🔴 Hard (2-4 problems)

- **Advanced Heap Problems**
  - Sliding window median
  - Employee free time

PATTERN 15: TRIE (PREFIX TREE)

Total Problems: ~15-20

🟢 Easy (8-10 problems)

- **Basic Trie Operations**
  - Implement Trie
  - Insert, search, delete in Trie
  - Count words with given prefix

🟡 Medium (5-7 problems)

- **Trie Applications**
  - Word search II
  - Replace words
  - Maximum XOR problems using Trie

🔴 Hard (2-3 problems)

- **Advanced Trie Problems**
  - Palindrome pairs

- Stream of characters

📈 OVERALL DIFFICULTY DISTRIBUTION

Difficulty	Count	Percentage
🟢 **Easy**	~180 problems	40%
🟡 **Medium**	~200 problems	45%
🔴 **Hard**	~70 problems	15%
Total	**~450 problems**	**100%**

🎯 RECOMMENDED STUDY PATH

Phase 1: Foundation (Weeks 1-4)

1. Basic Programming & Mathematics
2. Sorting Algorithms
3. Basic Array Operations

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

1. String Manipulation
2. Linked List Operations
3. Stack & Queue

Phase 3: Searching & Advanced Arrays (Weeks 9-12)

1. Binary Search
2. Advanced Array Patterns
3. Bit Manipulation

Phase 4: Tree & Graph Fundamentals (Weeks 13-16)

1. Tree Algorithms
2. Basic Graph Algorithms
3. Recursion & Backtracking

Phase 5: Advanced Topics (Weeks 17-20)

1. Dynamic Programming

2. Greedy Algorithms
3. Heap/Priority Queue
4. Trie

🔥 HIGH-FREQUENCY INTERVIEW PATTERNS

1. **Two Pointers** (40+ problems)
2. **Sliding Window** (25+ problems)
3. **Binary Search** (35+ problems)
4. **Tree Traversals** (30+ problems)
5. **Dynamic Programming** (50+ problems)
6. **Graph BFS/DFS** (25+ problems)
7. **Backtracking** (20+ problems)

This organization is based on the structure found in Striver's A2Z DSA Course repositories and typical problem distributions in comprehensive DSA curricula.