🚀 Programming Refresher: 20-Day Milestone-Based Study Plan

📋 Overview

- **Duration**: ~20 days (2-3 hours per day)
- **Structure**: 10 milestone-based checkpoints
- **Focus**: Core programming concepts through intermediate DSA
- **Based on**: Striver's A2Z DSA Sheet (Sheet-2)
- **Excludes**: Dynamic Programming (can be added later)

📚 Milestone Breakdown

Milestone 1: Refresh Core Programming Basics

Goal: Be comfortable writing small programs, loops, and functions again.

Topics Covered:

- Variables, data types, operators
- Conditionals, loops
- Functions & recursion basics

Practice Problems:

- Factorial calculation
- Prime number check
- Palindrome detection
- Reverse a string/array

Milestone 2: Arrays & Searching Foundations

Goal: Build fluency in handling arrays & searching techniques.

Topics Covered:

- Basic operations: traversal, min/max, reverse, rotate
- Sorting algorithms (bubble, selection, insertion)
- Binary search (1D + simple 2D)

Practice Problems:

- Two Sum
- Merge Sorted Array
- Search Insert Position
- Peak Element

Milestone 3: Strings & Hashing

Goal: Manipulate strings & use hashing for faster lookups.

Topics Covered:

- String operations (substring, anagrams, character frequency)
- HashMap & Set (duplicate detection, frequency maps)

Practice Problems:

- Valid Anagram
- Group Anagrams
- Longest Unique Substring

Milestone 4: Linked Lists

Goal: Understand and solve classic linked list interview problems.

Topics Covered:

- Singly linked list traversal, insert, delete
- Reverse a linked list (iterative + recursive)
- Detect cycle in a linked list

Practice Problems:

- Reverse Linked List
- Middle of Linked List
- Linked List Cycle

Milestone 5: Stacks, Queues & Sliding Window

Goal: Strengthen data structure usage for real-world problems.

Topics Covered:

- Stack basics (LIFO, valid parentheses, min stack)
- Queue basics (FIFO, circular queue, deque)
- Sliding window technique (fixed & variable size)

Practice Problems:

- Valid Parentheses
- Next Greater Element
- Maximum Subarray
- Longest Substring without Repeating Characters

Milestone 6: Recursion & Backtracking

Goal: Comfort with recursion patterns and basic backtracking.

Topics Covered:

- Recursive patterns (factorial, Fibonacci, subsets)
- Backtracking (generate permutations, subsets, n-queens basics)

Practice Problems:

- Subsets
- Permutations
- Generate Parentheses

Milestone 7: Bit Manipulation

Goal: Use bitwise tricks for elegant solutions.

Topics Covered:

- Bitwise operators & masks
- Common bit manipulation patterns

Practice Problems:

- Single Number
- Power of Two

- Number of 1 Bits

Milestone 8: Trees

Goal: Master tree basics & traversal techniques.

Topics Covered:

- Binary tree traversals (inorder, preorder, postorder)
- BFS / Level-order traversal
- Height & depth of a tree

Practice Problems:

- Maximum Depth of Binary Tree
- Symmetric Tree
- Binary Tree Level Order Traversal

Milestone 9: Graphs

Goal: Be able to handle graph basics and standard traversal.

Topics Covered:

- Representations (adjacency list, adjacency matrix)
- Graph traversals: BFS, DFS
- Connected components, island problems

Practice Problems:

- Number of Islands
- Flood Fill
- Clone Graph

🎯 How to Use This Plan

1. **Start with Milestone 1** and work through each topic thoroughly
2. **Practice the suggested problems** until you feel confident
3. **Don't rush** - if you need extra time on a milestone, take it

4. **Track your progress** by checking off completed topics
5. **Review weak areas** before moving to the next milestone
6. **Use Striver's A2Z DSA Sheet** for additional problems and detailed explanations

📝 Progress Tracking

Create a simple checklist for each milestone:

- [ ] Milestone 1: Core Programming Basics
- [ ] Milestone 2: Arrays & Searching
- [ ] Milestone 3: Strings & Hashing
- [ ] Milestone 4: Linked Lists
- [ ] Milestone 5: Stacks, Queues & Sliding Window
- [ ] Milestone 6: Recursion & Backtracking
- [ ] Milestone 7: Bit Manipulation
- [ ] Milestone 8: Trees
- [ ] Milestone 9: Graphs

🔗 Resources

- **Primary Resource**: Striver's A2Z DSA Sheet (Sheet-2)
- **Practice Platform**: LeetCode, HackerRank, or GeeksforGeeks
- **Time Allocation**: 2-3 hours per day