Codeforces Mastery Plan

A 42-Day Routine to Excel

Meraj Hossain Promit

July 24, 2025

Contents

1 Introduction & Philosophy

Welcome to your intensive Codeforces training plan! This guide is designed to provide structure and a clear path for the next 6 weeks (42 days). The goal is not just to solve problems, but to deeply understand the underlying algorithms and data structures.

1.1 Core Principles

- **Consistency over Intensity:** Solving 3-4 problems every day is better than solving 20 in one day and then burning out.
- **Deliberate Practice:** After failing to solve a problem for 30-45 minutes, read the editorial. Understand the solution completely, then code it yourself without copy-pasting.
- **Upsolving:** After every contest, spend time solving the problems you couldn't solve during the contest. This is crucial for growth.
- **Focus on A-E:** Problems A-C build speed and accuracy. Problems D and E introduce deeper algorithmic concepts. F-H are for very advanced contestants and are not covered in this plan.
- **Virtual Contests:** Participate in at least 2-3 virtual contests per week to simulate real contest pressure.

2 The 6-Week Plan

Each week focuses on a set of core topics. The problem list provides specific challenges for each day.

2.1 Week 1: Foundations - Math, Greedy, and Implementation

Focus: Strengthen your basics. These problems are often about careful implementation, observation, and basic mathematical concepts like number theory (GCD, LCM, primes) and combinatorics.

- **Monday-Wednesday:** Focus on Div. 2 A & B problems. Goal is speed and accuracy.
- **Thursday-Saturday:** Tackle Div. 2 C problems and easier Div. 2 D problems related to the week's topics.
- **Sunday:** Review the week's topics and upsolve or do a virtual contest.

2.2 Week 2: Sorting, Searching, and Two Pointers

Focus: Master binary search on the answer, sorting-based greedy solutions, and the two-pointers technique. These are incredibly common patterns.

- **Monday-Wednesday:** Solve problems that require sorting as a first step or use binary search.
- **Thursday-Saturday:** Focus on two-pointers and more complex binary search applications.
- **Sunday:** Review and participate in a virtual contest.

2.3 Week 3: Introduction to Data Structures

Focus: Learn and apply fundamental data structures.

- **Monday-Wednesday:** Stacks, Queues, Deques, Priority Queues.
- **Thursday-Saturday:** Maps, Sets, Fenwick Tree (BIT), and Segment Tree (basic range queries).
- **Sunday:** Review data structures and do a virtual contest.

2.4 Week 4: Graph Traversal and Basics

Focus: Build a solid understanding of graph theory.

- **Monday-Wednesday:** Depth First Search (DFS), Breadth First Search (BFS), connected components, cycle detection.
- **Thursday-Saturday:** Shortest path algorithms (Dijkstra), and Minimum Spanning Tree (Kruskal's).
- **Sunday:** Review graph algorithms and do a virtual contest focusing on graphs.

2.5 Week 5: Dynamic Programming I

Focus: Introduction to DP. The key is identifying states and transitions.

- **Monday-Wednesday:** 1D DP problems (coin change, basic path counting).
- **Thursday-Saturday:** 2D DP problems and DP with bitmasking on small subsets.
- **Sunday:** Review DP concepts and practice identifying DP subproblems.

2.6 Week 6: Dynamic Programming II & Mixed Review

Focus: More advanced DP and combining topics from previous weeks.

- **Monday-Wednesday:** DP on trees, Digit DP.
- **Thursday-Saturday:** Mixed practice. Solve random problems from recent contests (Div. 2 C, D, E) to practice identifying the topic yourself.
- **Sunday:** Final review and a full virtual contest to gauge your progress.

3 Problem List

Here is a list of **126 problems** to guide your daily practice. Links are clickable in the generated PDF.

Week	**Day**	**Problem (ID)**	**Key Topic(s)**
1	Mon	1328A Divisibility Problem	Math, Implementation
		1352A Sum of Round Numbers	Math, Implementation
		734A Anton and Danik	Ad-hoc, Implementation
	Tue	1360B Honest Coach	Sorting, Greedy
		467A George and Accommodation	Implementation
		1512A Spy Detected!	Arrays, Implementation
	Wed	1374C Move Brackets	Greedy, Stack idea
		1353B Two Arrays And Swaps	Greedy, Sorting
		1475B New Year's Number	Math, DP idea
	Thu	1324C Frog Jumps	Greedy, Implementation
		1367B Even Array	Constructive, Implementation
		1409B Minimum Product	Greedy, Math
	Fri	1335C Two Teams Composing	Greedy, Frequency Count
		1294C Product of Three Numbers	Number Theory, Factorization
		1490D Permutation Transformation	Recursion, Constructive
	Sat	455A Boredom	DP (intro), Frequency Count
		279B Books	Two Pointers / Binary Search
		1399C Boats Competition	Two Pointers, Hashing
2	Mon	1352C K-th Not Divisible by n	Binary Search, Math
		165C Another Problem on Strings	Hashing, Two Pointers
		706B Interesting drink	Binary Search
	Tue	1475C Ball in Berland	Combinatorics, Hashing
		474B Worms	Binary Search, Prefix Sums
		1324D Pairs	Binary Search, Two Pointers
	Wed	1343C Alternating Subsequence	Greedy, Two Pointers
		1201B Zero Array	Math, Greedy
		363B Fence	Sliding Window, Prefix Sums
	Thu	600B Queries about less or equal elements	Two Pointers
		271B Prime Matrix	Number Theory, Sieve
		1154B Make Them Equal	Ad-hoc, Set
	Fri	1520E Arranging The Sheep	Greedy, Median idea
		1462D Add to Neighbour and Remove	Greedy, Prefix Sums
		1006C Three Parts of the Array	Two Pointers

Continued on next page

Table 1 continued from previous page

Week	Day	Problem (ID)	Key Topic(s)
	Sat	1426D Non-zero Segments 492B Vanya and Lanterns 1370C Number Game	Prefix Sums, Hashing Binary Search, Sorting Game Theory, Math
3	Mon	5C Longest Regular Bracket Sequence 1490E Accidental Victory 4C Registration system	Stack, DP Greedy, Sorting Map (Hashing)
	Tue	1141B Maximal Continuous Rest 1327B Princesses and Princes 1363A Odd Selection	Implementation, Two Pointers Greedy Combinatorics, Greedy
	Wed	977C Less or Equal 1203B Equal Rectangles 1703E Mirror Grid	Sorting Sorting, Hashing Implementation, Matrix
	Thu	339D Xenia and Bit Operations 61E Enemy is weak 1472D Even-Odd Game	Segment Tree Fenwick Tree (BIT) Game Theory, Greedy
	Fri	1108D Diverse Garland 1359B New Theatre Square 1551C Interesting Story	DP, Greedy Greedy Greedy, Custom Sort
	Sat	1692F 3SUM 1538C Number of Pairs 816B Karen and Coffee	Hashing, Implementation Two Pointers Prefix Sums (Difference Array)
4	Mon	115A Party 580C Kefa and Park 520B Two Buttons	DFS, DSU DFS/BFS on Trees BFS, Greedy
	Tue	20C Dijkstra? 977D Divide by two and multiply by three 1020B Badge	Dijkstra Graph, Sorting, Backtracking DFS, Cycle Detection
	Wed	1201C Maximum Median 1092B Teams Forming 429B Working out	Binary Search, Greedy Sorting DP on Grids
	Thu	427C Checkposts 500A New Year Transportation 1385D a-Good String	Tarjan's (SCC), Graphs Graph Traversal (DFS/BFS) Recursion, Divide and Conquer
	Fri	1335D Anti-Sudoku 228A Is your horseshoe on the other hoof? 1360E Polygon	Constructive, Implementation Set, Implementation DP, Implementation
	Sat	771A Bear and Friendship Condition 602B Approximating a Constant Range	DSU/DFS Two Pointers, Data Structures

Continued on next page

Table 1 continued from previous page

Week	Day	Problem (ID)	Key Topic(s)
		1398C Substring Frequency	Hashing, Prefix Sums
5	Mon	118D Caesar's Legions 706C Hard problem 1097B Petr and a Combination Lock	DP DP, Strings DP, Brute Force, Bitmask
	Tue	474D Flowers 1420C2 Pokémon Army (hard version) 414B Mashmokh and ACM	DP, Prefix Sums DP, Greedy DP, Number Theory
	Wed	166E Tetrahedron 545C Woodcutters 1155C Alarm Clocks	DP (Matrix Exponentiation) DP, Greedy Math, GCD
	Thu	988D Points and Powers of Two 1324E Sleeping Schedule 1350B Orac and Models	Brute Force, Hashing DP DP
	Fri	839C Journey 1036C Classy Numbers 1454D Number into Sequence	DFS, Probability Digit DP Number Theory
	Sat	131D Subway 1133D Zero Quantity Maximization 466C Number of Ways	BFS (Multi-source) Math, Hashing Prefix Sums, Combinatorics
6	Mon	1325C Ehab and Path-etic MEXs 1469B Red and Blue 327A Flipping Game	Trees, Greedy Greedy, Prefix Sums Brute Force, DP
	Tue	1526C2 Potions (Hard Version) 1328D Carousel 915B Browser	Greedy, Priority Queue Constructive, Greedy Implementation, Math
	Wed	1428C ABBB 295B Greg and Graph 1348B Phoenix and Beauty	Greedy, Stack Floyd-Warshall (Unusual) Constructive, Periodicity
	Thu	1499C Minimum Grid Path 1519D Maximum Sum of Products 489C Given Length and Sum of Digits...	Greedy, Prefix Sums DP, Optimization Greedy, Constructive
	Fri	1511D Min Cost String 1506E Restoring the Permutation 1285C Fadi and LCM	Constructive, Graphs Greedy, Data Structures Number Theory, Brute Force
	Sat	1535D Playoff 1547E Air Conditioners 148A Insomnia cure	DP on Trees, Implementation DP, Optimization Math, Implementation

4 Final Advice

- **Don't get stuck:** It is better to learn the solution from the editorial and move on than to be stuck on one problem for a whole day.
- **Code Quality:** Write clean, readable code. This helps in debugging and modifying solutions under pressure.
- **Templates:** Prepare a code template for contests. It should include common headers, macros, and functions for fast I/O.
- **Rest:** Don't forget to rest. A tired mind cannot solve problems effectively. Get enough sleep, especially before a contest.

Good luck on your Codeforces journey!