Competitive Programming

Starting in 5 mint

DSA vs CP

Dev vs CP

Benefits

- Edge in Resume shortlisting & Referrals (TIER 3)

Practice

- Problem solving ability ↑↑ → hot of help in DSA
- Logical thinking ↑↑
- Online coding Round → Cakewalk (TIER 1/2)
  - → Pressure of time bound contest
- Interviews → Expectation ↑↑
  - SP Salesforce

Pre-requisite

#language → C++ + STL → My notes

→ Java + Collection frameworks → Java → backend

→ OOPS

Why to prefer C++ over Java?

→ Faster Execution

→ Editorials → C++

→ Standard Template library

→ Less verbose

Mathematics

1. Logarithms
2. Permutations & Combinations
3. Sequence & Series (AP & GP)
4. Basics of Geometry
5. Number Theory & Combinatorics
   - GCD & LCM (Euclid)
   - Prime No & Factorization
   - Inclusion Exclusion principle

DSA → Graph / DP (C&D)

6. Bit manipulation
7. Modular Arithmetic → 10^9+7
8. Game Theory

Setting up

Code Editor :→ Sublime Text (for C++)

- → Starter Code
- → Fast Input
- → Codes Snippets
  - typedefs
  - modulo (10^9+7)
  - Bitwise operations
  - Segment Trees, Number theory, etc.
- → Keyboard shortcuts
  - ↳ Typing speed ↑↑

https://www.youtube.com/watch?v=8SPJHYDGIu0

https://www.youtube.com/watch?v=Mt6Jb8u9XBk

} Chaitanya (6*)

Articles → Geeks for Geeks
→ codeforces
→ cp-algorithms

Extra Problems → CSES (2nd) 2 ans
→ Atcoder DP Set (25)

Video Resources

① NADOS level ② & ③

② Youtube channels → luv
→ Errichto
→ Utkarsh Gupta } Sp - Expert

Competitive Programming

→ Pick one platform & solve every live contest

Websites

- ⭐ Leetcode (PSA)
  - → Short
  - → long X
- Codechef
- ⭐ Codeforces
- Atcoder

Big Competitions

GOOGLE

→ Hash code { in group of 2-4 } → Hackathon (21 Feb)

→ Code Jam { 3-4 rounds } → Codeforces Div 1

→ Kick Start { monthly 2-3 hr contests }

↳ Leetcode & Codeforces Div 2

FACEBOOK → Hackercup → Codeforces Div 1

ACM ICPC → for Candidate master & above! (red)

Codeforces

how to approach a problem?

→ Extract WHAT from a story.
↳ Examples ⭐

→ Try to figure out Which DS or Algorithm to apply

→ Check time constraint & code only if Time Complexity will satisfy.

→ Figure out corner cases ← trivial cases/boundary cases
- Integer overflow
- TLE
- Stack overflow

→ Submit only if 90% sure!
{ Dry run on 2-3 test cases }

Suggestion: → Check submission & acceptance count before trying

Div 1 vs Div 2 vs Div 3

Div 1 → Expert onwards

Div 2 → below candidate master

Div 3 → below expert

Educational Contests

↳ Problems weightage equal

Sample Test cases ≠ System Test cases

UPSOLVING → Reading editorial part by part

Pair programming

Virtual contests (Unrated)

2 ques 10E

3 ques 20M

30 4