≪Coding Club≫

COMPETITIVE PROGRAMMING

INTRODUCTION TO COMPETITIVE PROGRAMMING

- Competitive programming is a mind sport where participants solve well-defined computational problems within a limited timeframe.
- The goal is to write efficient algorithms to solve problems correctly and quickly. It's widely used to enhance problem-solving skills, coding proficiency, and algorithmic knowledge.

PLATFORMS AND CONTESTS

There are numerous online platforms that host competitive programming contests, such as Codeforces, LeetCode, HackerRank, and CodeChef. These platforms provide a wide range of problems and regularly organize contests where participants can compete against each other and climb the rankings.

SKILLS DEVELOPMENT

Engaging in competitive programming helps improve various skills, including algorithmic thinking, data structures, coding efficiency, and debugging. It also enhances the ability to work under pressure and develop innovative solutions to complex problems.

USEFUL LINKS

- https://neetcode.io/roadmap
- https://takeuforward.org/strivers-a2z-dsa-course/strivers-a2z-dsa-course-sheet-2
- https://www.techinterviewhandbook.org/grind75?weeks=28&hours=6
- https://github.com/DeathStroke19891/CP_Notes

DYNAMMIC PROGRAMMING

- Identify the subproblems: Determine how to break the main problem into smaller, manageable subproblems.
- Formulate the recurrence relation: Find a way to express the solution of the problem in terms of the solutions to its subproblems.

DYNAMMIC PROGRAMMING

- Choose a DP approach:
- Top-Down (Memoization): Solve the problem recursively and store results of subproblems to avoid redundant calculations. Higher Space and Time Complexity, Easier to visualize.
- Bottom-Up (Tabulation): Solve the problem iteratively, starting from the smallest subproblems and building up to the solution of the main problem. Lower Space and Time Complexity, Harder to visualize.
- Implement the DP solution: Write the code to implement the chosen approach, ensuring that subproblem results are reused efficiently.

DYNAMMIC PROGRAMMING

746. Min Cost Climbing Stairs

DYNAMMIC PROGRAMMING

198. House Robber

DYNAMMIC PROGRAMMING

213. House Robber II

DYNAMMIC PROGRAMMING

https://github.com/DeathStroke19891/EC_Skill_Lab_Resources

RASHTREEYA SIKSHANA SAMITHI TRUST
RV
INSTITUTIONS

≪Coding Club≫

THANK YOU