COMPETITIVE PROGRAMMING?

Competitive Programming is in the air at the moment. Here I present before you what hype is all about, this blog will be tackling basic questions like, “what is CP?”,why CP?” And how can we get started with it? And most importantly Is CP dead?,e.g. So buckle up & let’s get started with this journey of knowing about Competitive Programming from basics.

A thorough comprehension of algorithms and data structures, as well as the capacity for logical thought and problem-solving, are essential skills for competitive programming. Participating in these programming competitions gives people the chance to compare their abilities to those of their colleagues and gain knowledge from others. It aids in the development of problem-solving abilities, which are beneficial in a variety of professions, including research, data science, and software development.

Let's talk about the fundamentals and advantages of competitive programming, including the kinds of challenges that are often encountered in competitions and some well-liked programming tools like online courses, given its potential for skill development. This blog will offer helpful information whether you are a novice trying to get started or an experienced coder looking to advance your abilities.

Competitive programming requires a solid understanding of algorithms and data structures as well as the ability to reason logically and solve problems. People have the opportunity to compare their skills to those of their peers and learn from others by taking part in these programming competitions. It supports the growth of problem-solving skills, which are advantageous in many different vocations.

Let's discuss the foundations and benefits of competitive programming, as well as the types of difficulties that are frequently experienced in contests and some popular programming resources like online tutorials, given its potential for skill growth.

- This puts a time constraint on contenders' ability to solve algorithmic puzzles. This enhances one's capacity for problem-solving as well as for writing effective, optimised code. Additionally, it fosters a thorough understanding of data structures and algorithms.
- Competitive programming is a common recruiting strategy used by businesses. Additionally, competitive programming can help candidates in technical interviews by enhancing their abilities and expertise. Additionally, it gives participants a preview of the kinds of inquiries that will probably be made during these interviews.

- Competitive programming requires contestants to think critically and creatively, and this can help improve overall thinking ability. It also helps to develop the ability to analyze and solve problems under pressure.
- This is a fantastic place to begin learning programming. Along with building a solid basis in algorithms and data structures, it also gives the user a sense of achievement and fulfilment.

Competitive programming problems can be found on online platforms such as CodeForces, HackerRank, LeetCode, SPOJ, and CodeChef, among others. They are often used in university classes, coding competitions, and as a way to assess job applicants. Some popular problem types include:

1. Searching

This type of problem involves searching for a specific element or pattern in a given data set.

2. Sorting

These entail sorting an array or list of elements in a specific order.

3. Dynamic Programming

This requires breaking down a problem into smaller subproblems and solving them in a specific order to optimize the overall solution.

4. Graph Theory

These problems involve manipulating and analyzing graphs, such as finding the shortest path between two nodes, or determining if a graph is connected.

5. Number Theory

This encompasses mathematical concepts and operations, such as prime numbers and modular arithmetic.

6. Greedy Algorithm

These focus on making the locally optimal choice at each stage with the hope of finding a global optimum.

7. Backtracking

Participants have to try out all possible solutions to these problems and discard the ones that don’t work.

8. Divide and Conquer

These problems need to be broken down into smaller subproblems that can be solved independently, after which the solutions are combined to solve the original problem.

9. String Manipulation

This problem-solving involves working with strings, with actions such as finding the length of a string, reversing a string, or finding a substring.

10. Data Structure

To solve these problems, you have to work with various data structures, such as stacks, queues, linked lists, trees, and graphs.

Starting off in competitive programming can be challenging, but with these pointers you can advance steadily:

- Study the fundamentals of programming.
- Learn the intricacies of the programming language of your choice and begin tackling issues on websites like LeetCode, HackerRank, CodeForces, CodeChef, and SPOJ.
- Learn how to solve problems, and comprehend the time and space complexities of various algorithms.
- To enhance time management, practise frequently and take part in online tournaments.
- Join a group to get advice from others.
- Track your development and pinpoint specific areas for growth.
- Enjoy yourself and don't give up if you run into trouble.

Recently, Google stopped hosting coding contests, which sparked a debate in the community about whether or not CP was still alive. I think I've summed up my thoughts here. Is CP dead, you ask? A. Despite Google ending its Coding Competitions, CP is still alive and well. While there are still many more online platforms and competitions available for competitive programming lovers, Google's Coding Competitions gave programmers a place to compete and demonstrate their abilities.

Codeforces, Topcoder, Atcoder, LeetCode, HackerRank, and CodeChef are a few examples of well-known online CP platforms. These websites frequently hold coding competitions, giving participants a chance to sharpen their abilities and take on other programmers from across the globe.

Furthermore, CP has been more well-known in recent years as a crucial competency for technical interviews in the software business. Thus, it is doubtful that the demand for competitive programming talents will go down anytime soon. Thus, even though Google's Coding Competitions are no longer being held, the CP community still offers plenty of chances for programmers to share their talents and pick up new ones.

Now coming to the end of this blog, I would like to conclude by mentioning that, **Competitive programming is a brain-sport**. As you start solving harder and harder problems, your analytical and rational thinking keeps on intensifying.