SKILLS

COMPETITIVE PROGRAMMING
What beginners should know

SADMAN AHMED

At present, competitive programming is gaining popularity everywhere, and it's not only limited to Computer Science students either. But between all the noise, it is hard for a beginner to navigate the way towards excelling in competitive programming, and there are several misconceptions too.

One major misconception surrounding competitive programming is that it enhances one's software development skills. While it is true that competitive programming increases problem-solving skills, analytical and critical abilities, and logical thinking, it has almost little to no relation with software development. On that note, competitive programming has less to do with the industry and more with your own preparation for the industry.

The bottom line is that competitive programming is a specific skill set similar to many other skill sets, and it is incorrect to assume that being an expert in it would automatically result in being an expert in other unrelated fields.

Another issue that arises for a beginner is what to learn and how to learn. The programmer becomes torn between mastering data structure and algorithms, and

ILLUSTRATION: **FATIMA JAHAN ENA**

> Going one step further, the programmer should look for solutions even if they can solve the problem alone. Because a problem is solvable in multiple ways, analysing other solutions can help one find a new perspective which would be better in the longer run. The goal here is to learn, and as long as that goal is being served, it is okay to check out existing solutions.

solving problems. It creates a paradoxical scenario, where a programmer might learn all the relevant data structures and algorithms but can't solve problems due to inexperience. But on the other hand, if they try to solve problems, they need the knowledge of data structures and algorithms.

To solve this issue, learning on the go is a better approach. It is the practice of only learning something when it is needed. The programmer should try to solve problems based on the knowledge they have already gained at first, and if it isn't possible at all, then only learn new techniques. This approach, while not

without its drawbacks, balances the issue of learning versus solving.

Seeking help and looking at solutions is a major issue many competitive programmers have to deal with. Being an expert in competitive programming requires a lot of time and effort, and no one can learn everything all by themselves. Seeking help from other programmers, seniors, the programming community in the respective institution or online forums like Stack Overflow is very important. Stack Overflow resources are quite handy, as the forum has been running for a long time. The same goes for looking at solutions online. In the beginning, programmers have a hard time looking at solutions because of the mindset of "Why should I look for solutions? Isn't it a problem that I can solve without looking?" But this is counterproductive for several reasons.

When a programmer is stuck with a problem for a long time, not looking at the solution and keeping on trying is unnecessary and wastes time. Checking the solution and taking it as a learning experience is more fruitful because understanding others' codes is also a valuable skill, and the programmer can use the knowledge of the solutions in later problems.

Going one step further, the programmer should look for solutions even if they can solve the problem alone. Because a problem is solvable in multiple ways, analysing other solutions can help one find a new perspective which would be better in the longer run. The goal here is to learn, and as long as that goal is

being served, it is okay to check out existing solutions.

When getting into competitive programming, there is no hard-and-fast rule to start solving only difficult problems. One can and probably should start with the basic problems first on sites like CodeChef, CodeForces, Hackerrank, etc. Sites like TopCoder or LeetCode are much more difficult for beginners, but solving problems from there can still be a valuable experience, especially LeetCode.

Motivation is another important factor for beginners. Competitive programming, like any other competitive activity, is highly stressful. At the same time, it also has its unique problems. Searching for a solution and not being able to find it for days, being unable to solve even a single problem in a competition, and the anxiety of falling behind others are all part of being a competitive programmer. The only way to go through this is to enjoy the art of programming because, without motivation, it is very easy to burn out.

Competitive programming can be a rewarding practice. Those who have had the joy of solving a problem after a lot of brainstorming can attest to that. Even after following all the guidelines, one might face unique difficulties, and they have to solve them on their own. That is where the real satisfaction lies, which is something I wish every programmer can experience.

Sadman Ahmed Siam is a student at Islamic University of Technology (IUT).