

Closing Remarks

We'll cover the following ^

- Participate in contests
- Upsolve
- Starting out
- Sample contest
- What's next

Participate in contests

Hopefully, I have provided you with enough resources to start competing in programming competitions.

Typically, such contests last for a few hours (2-5) with about five questions in a short contest, and up to 12 or more in longer contests. The questions gradually get harder, require more data structure and algorithms knowledge, and of course practice.

Upsolve

Upsolving is a very common term among competitive programmers. Basically, the idea is this: when you participate in a contest, say you solved 3 out of 6 problems. You should *always* solve the next couple of questions after the contest; look at the tutorials, other participant's code, whatever it takes.

See each problem as a chance to learn something new. Find out what you didn't know in order to solve that problem. Read about it, spend hours or even days if needed, to understand the solution.













This is the best thing you can do to improve every day.

Starting out

Once you start participating in solving problems, you will immediately see that you can solve the first few easiest problems with what we have covered so far.

Sample contest

Let's see what a two hour contest on codeforces.com looks like to give you a general idea.

Problems				
#	Name			
A	Suffix Three	standard input/output 1 s, 256 MB		 x9822
B	Azamon Web Services	standard input/output 2 s, 256 MB		 x4196
C	Cut and Paste	standard input/output 2 s, 256 MB		 x1775
D	Beingawesomeism	standard input/output 2 s, 256 MB		 x1200
E	Jeremy Bearimy	standard input/output 3 s, 256 MB		 x335
F	Miss Punyverse	standard input/output 4 s, 512 MB		 x38

Problems are arranged by difficulty level here, and you can see the number of successful submissions on the right-hand side.

The first few problems are generally ad-hoc, little logic, basic maths, simple DS and Algos.

As you move to medium problems, you will need to understand the theory and application of complex data structures like graphs, segment trees, etc.

The hardest problems are generally a combination of multiple concepts and good observations and even more complex topics.

What's next

The idea is not to cover everything theoretically in this course, but to ease new people into competitive programming as it can be overwhelming without a solid foundation.

In the next level(s) of this course, I intend to cover harder topics, implementation tips, problem-solving techniques, and more. Everything at this level is a pre-requisite for the next level(s).

Requirements for the next level(s):

So, I would suggest to start participating in contests right away to master these topics.

You have thus leveled up.

All the best and I will see you at the next level!