Tips and Tricks for Competitive Programmers | Set 1 (For Beginners)

Difficulty Level : Easy • Last Updated : 21 Jun, 2018

This article is a collection of various tips that would help beginners of Competitive programming to get an insight of things that should or shouldn't be done.

Competitive programming can only be improved by "PRACTICE, PRACTICE AND PRACTICE". Try to solve as many questions you can solve on sites like practice.geeksforgeeks.org. This will enhance your mind to think more on algorithms.

Start with the beginner section, and when you feel comfortable with that, move on to higher level i.e. easy, medium, and hard and so on. Try to attempt all the questions yourself and don't see the solution before attempting it. Don't feel demotivated when you get wrong answers, it's just part of the learning. The more you practice the more you learn. Just be passionate about coding and practice.

Few Days before you begin:

1. **Learn -Practice-Repeat** -Try to learn a new concept on a daily basis. Solve questions daily, one or two if not more!! After going through a new algorithm or technique, we should immediately search for its applications and attempt problems. Like if you learn dynamic

programming, try to finish up all its problems. Adapt the habit of reading which most of the youngsters don't have nowadays.

2. **Write before coding**– Implement all algorithms yourself rather than copying from someone else. Make yourself written notes while studying these concepts. Mathematics is great area to start competitive programming.
3. **Getting Edgy-** During practice always solve that problem that is just at the edge of your knowledge i.e., you don't exactly know how to solve the problem but you know what you should know to solve that problem. For example, you look at the problem and you can tell that it's a simple graph problem but you do not know anything about graph.
4. Trees, Graphs, Algos– Make sure you are thorough with the concepts of trees, graphs and important algorithms as there is at least one question making use of their applications in every contest or a company hiring round.
5. **Short is sweet**– Long contest is good for learning but try to take part in more and more short contests. Short contest is the real competitive programming. We should make it a must habit to spend some short time during peak hours in a programming forum where top coders usually hangout sharing their insights and often get into discussions.
6. **Complexity is Complex**– Do not be obsessed with lower and lower execution time. Do not waste time on over-optimizing your solution. If the solution is accepted, move on to next problem. First just get into the habit of coding daily and then worry about complexities.
7. **Hard must come**– Some people say Stick to one website for practice while others believe you must taste all bunches. Whatever you decide , slowly but surely start solving harder problems.
8. **Target Job**– If you are frequently participating in the contests which are meant for jobs then make sure to read all previous questions, algorithms and related stuff to cut short your efforts as well as

selection time.You must attempt previous company questions too.

Last day before the contest especially If you are attempting the contest for Getting Hired

9. Don't look for new problems because that may create panic in your mind.
10. Take sufficient amount of sleep the night before. Keep your mind relaxed and stay stress free

During the contest

1. **Be Attentive** – Most of the programmers when see a new question, they will hurry in typing it on system before pre planning or before writing logic to crack that task. Sometimes they will stick at a point in between of typing code in system and might need to start coding again. If we avoid typing in system before cracking the logic it will be helpful to save time. One should start with:
    1. Reading the problem statement at least twice
    2. Analyzing the problem statement
    3. Input output pattern should be kept in mind before submission

and read problem many times to understand concept behind problem.
4. Use pen and paper to develop the logic and then code
5. Read the instructions of the contest carefully (**Time limit, Meaning of various symbols used in the contest page, Number of submissions allowed etc.**)

2. **Clock is ticking**– Keep an eye on the clock .If you are unable to solve a particular question, you very well have the option to go to next.
3. **Test the test cases**– If your code is not accepted, then go through your code again and check about variable declaration, complexity of the code, and try checking your code for multiple numbers of test cases.

After the contest

1. **Editorials are MUST**– After submission; even if your code is accepted then just don't jump on to the next question. Try to read editorial of the question , this will help you know better and efficient solution of that question.
2. **Geeks around may know better-** Examining codes written by other eminent coders will reveal great insights (if it is allowed). Even reviewing other's solution to a problem we have solved might expose some of the unique features of the problem and aids us in viewing the same problem from a different point of view. The important point here is that- you may come across different algorithms that are used to solve questions, learn those algorithms and make sure that you understand them.
3. **Practice**– Don't worry if you are unable to solve the questions there, it just means that you need more practice.
4. **Past teaches future**– It is a good practice to stick to a problem which we are unable to solve for at least 2 days. On reviewing the solution, we will understand where we deviated from the correct path and

would aid our thinking process in future attempts. We should make a note of the problems which we were unable to solve and hence, we went for the solution. We should make sure to review the same problem after a couple of weeks and attempt to solve it entirely.

5. **Time is precious**– Preparation for anything is very important be it for a contest day, an exams or a project submission, which student mostly fail to do. Preparing at the last moment often fail short of the expectations. Give enough time to go through algorithms, sample problems and work upon your own strengths and weaknesses.

Happy Coding!!

This article is exclusively drafted by contributions of our Campus Geeks- **Rahul Agarwal, Aditya Chatterjee, Shubham Singh Rajput, Vineet Sethia, Saiteja Reddy, Shaily Seth, Mudit Maheshwari and Ajay Jain.**

Tips and Tricks for Competitive Programmers | Set 2 (Language to be used for Competitive Programming)

If you like GeeksforGeeks and would like to contribute, you can also write an article and mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

Previous

Tips and Tricks for Competitive Programmers | Set 2 (Language to be used for Competitive Programming)

Next

Resume Writing For Internship

RECOMMENDED ARTICLES

Page : 1 2 3

programming (for C++ 11)
03, Jun 17

Programming
07, Jun 17

Article Contributed By :

GeeksforGeeks

Vote for difficulty

Current difficulty : Easy

Easy | Normal | Medium | Hard | Expert

Article Tags : Competitive Programming

Improve Article | Report Issue

Load Comments

GeeksforGeeks
5th Floor, A-118,
Sector-136, Noida, Uttar Pradesh - 201305

feedback@geeksforgeeks.org