What is the path starting from 0 in programming to get very good at competitive programming?

Answer · Follow · 33 · Request

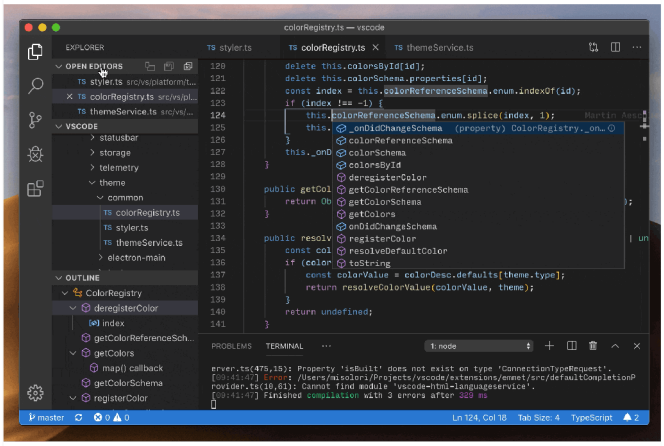
All related (37) · Sort · Recommended

Meet Joshi · Following
Gold Medalist, IEEEXtreme 13.0, India · Updated 1y

Different people might have a different perception of starting from 0 to get very good at competitive programming. I respect all their opinions. Today, I would be sharing my point of view. You might **pick-up the points that you find helpful**.

Please follow this order sequentially

- **Follow the following topics on Quora:**
  - Competitive Programming
  - Algorithms
  - Computer Programming
- **Subscribe the following YouTube channels:**
  - Tushar Roy - Coding Made Simple
  - take U forward
  - WilliamFiset
  - Aditya Verma
  - Pepcoding
  - Code NCode
  - mycodeschool
  - Gaurav Sen
  - Rachit Jain
  - code_report
  - Abdul Bari
- **Study the style guide before starting to write the code.**
  It ensures excellent and clean coding practice. Always keep it along with you while writing the code, or memorize it. If you skip this step, later on, you will learn it the hard way. Pick any one of the guides and stick with it.
  - C++ Style Guide: by Google or The C++ Core Guidelines
  - Java: by Google
  - Python: by Google
  - or you can choose your own, with which you are comfortable.
- **Install Visual Studio Code**. It is completely free and one of the best IDEs that I have come across. Most high rated programmers code in this IDE. It has lots of custom features, specifically carved for programmers.
  Refer to online resources for its installation and usage guidelines.



- **Complete the free GeeksForGeeks Fork series** depending on your language of preference. If you code in C++, I recommend you to complete both C and C++ tracks.

  Links
  - Course | Fundamentals of Programming with C
  - Course | Fork CPP
  - Course | Fork Java
  - Course | Fork Python

  The time required: 1-month per course

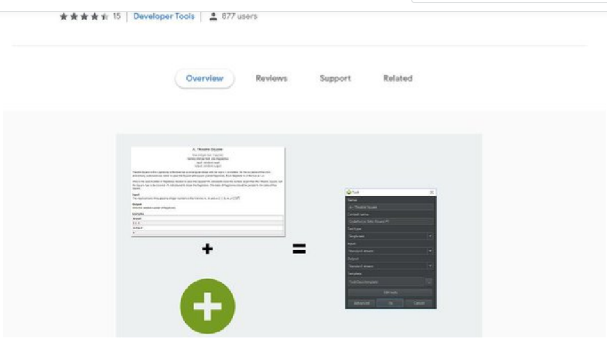
- Install **Competitive Companion Google Chrome extension** and learn how to use it.

Upvote · 1.8K · 29 · 144

Related questions

- What is competitive programming?
- What is the best strategy to improve my skills in competitive programming in C++ in 2-3 months?
- What is the best way to start competitive programming?
- What is the best coding competition site for beginners?
- What have you gained from competitive programming? Did you go into research? Did it help you in any aspect as a software engineer? Did it help you get an in-depth knowledge of a programming language? Did it affect your problem-solving skills?
- How can I improve my logical skills for programming?

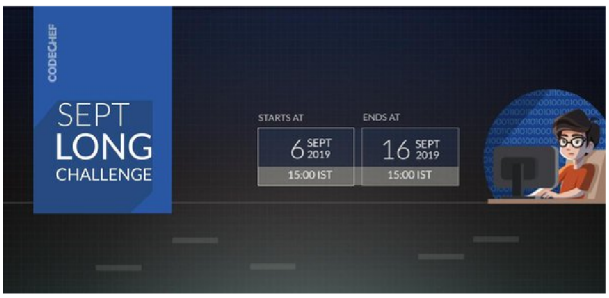
Add question



- Learn to **prepare your custom template and snippets**. I recommend storing your custom snippets on G-Drive, as MS-Visual Code does not have a backup and restore feature.
  NOTE: As you have done some basic coding, you must be familiar with the syntax of the language of your preference. **I recommend you to make a custom template**, rather than copy-pasting from somewhere.

```
1
```

- **Start participating in Codechef long challenges and unrated contests.**
  Please don't jump into rated contents apart from Codechef long; it might hamper your momentum.

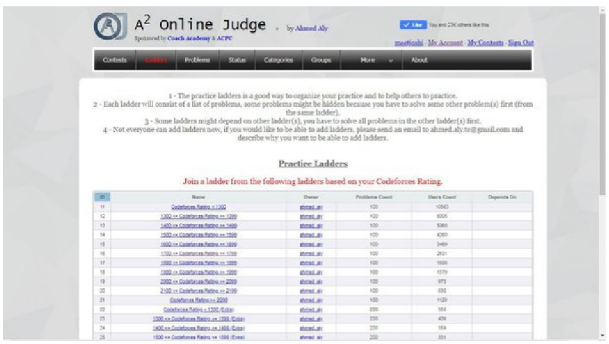


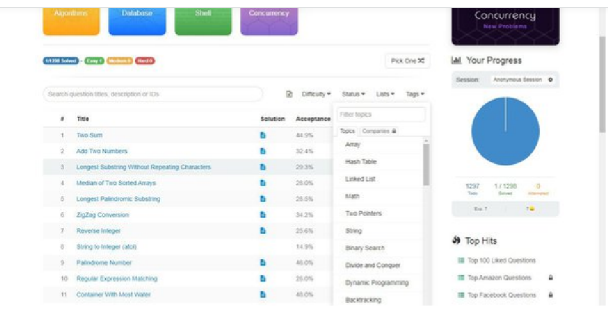
- Parallelly, solve questions from any one of the following platforms:
  - **CodeChef Beginner**

Beginner | Easy | Medium | Hard | Challenge | Peer

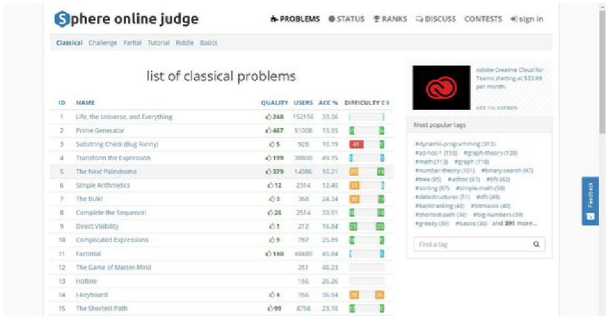
Name	Code	Successful Submission	Accuracy
ATM	HS08TEST	90688	19.96
Enormous Input Test	INTEST	71138	44.57
Add Two Numbers	FLOW001	41471	53.91
Small factorials	FCTRL2	36345	18.52
Sum of Digits	FLOW006	32850	49.97
Turbo Sort	TSORT	32524	25.43
Number Mirror	START01	31404	69.47
Find Remainder	FLOW002	30672	62.42
The Lead Game	TLG	26986	21.41
First and Last Digit	FLOW004	24704	57.54
Lucky Four	LUCKFOUR	24495	63.42
Packaging Cupcakes	MUFFINS3	24354	51.85

  - **Codeforces Ladder**

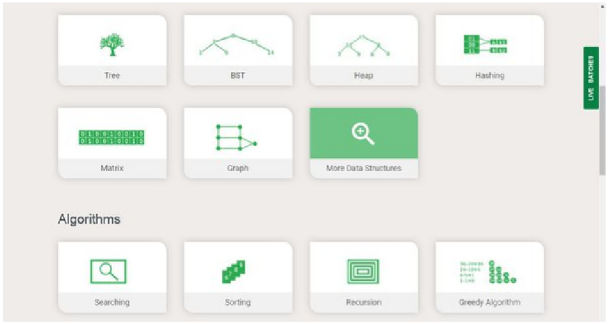




- **SPOJ**

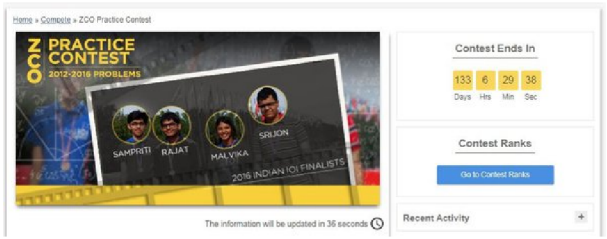


- **GeeksForGeeks Practice**

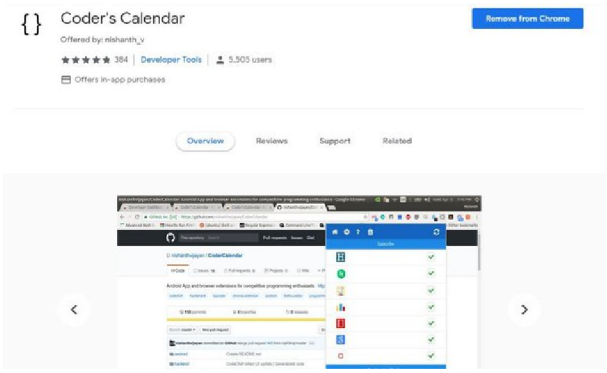


  - **Solve at least 100 of them** before progressing further.
    You will get familiar with the **coding environment, IDEs, discussion forums, tags, editorials, etc**. Also, you will get a basic idea about *input-output format, data types, memory issues, overflow issues, modulus handling, time complexity, different types of errors (RUNTIME, Compilation, TLE, etc.) and best practices in coding*.
    Refer to editorials as a last resort. Try solving questions on your own.
    The time required: 120 days (at max)

- **Participate in the ZCO practice contest** and try to solve as many questions as possible.
  This contest is for school students, so knowing this, you might get a headstart. There are lots of blogs and tutorials available online for each of these questions.
  The time required: 1 month (at max)

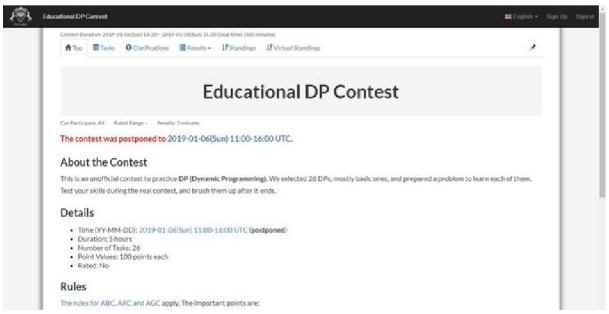


- Now, you are all set to participate in rated contests. **Participate in as many contests as possible**. Participating in contests help you solve more questions in less time. **Add this extension** and choose your favorite platforms.
  NOTE: Always Upsolve

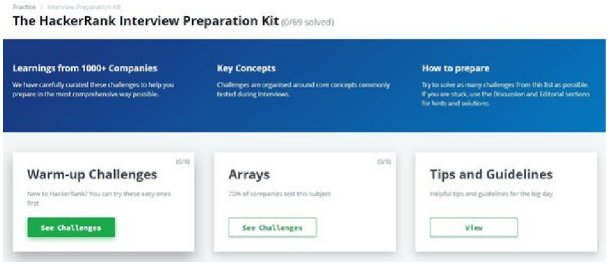


- **Register on InterviewBit and complete all levels.** There are around 300 questions in total; solving all of them will make you a master of fundamental algorithms. The best part is, everything is free.
  The time required: 6 months (at max)

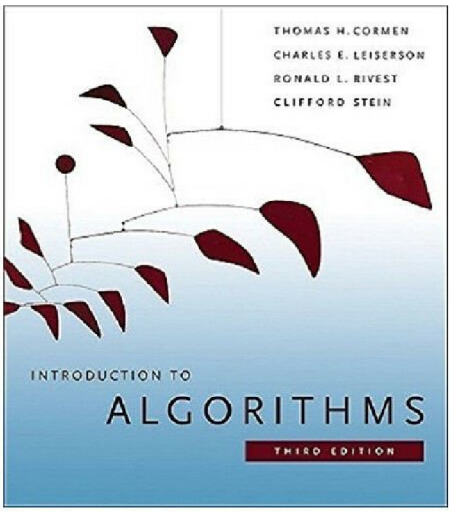
- Participate and Solve all the questions of the **AtCoder DP Contest.** It will make you a master of Dynamic Programming. The Contest covers every nook and cranny of Dynamic Programming questions. No question of DP can be of the concept out of this contest.



- **Complete the HackerRank Interview Preparation Kit.** It is a collection of 69 questions arranged topic-wise.
  The time required: 3 months (at max)



- **Read CLRS.**
  Softcopy, solutions, everything is available online for free. Also, lots of resources related to that are available.
  The time required: while(1) {repeat;}
  This book is the bible of competitive programming. It never gets old. The more you read, the better you get.



KINDLY READ DESCRIPTION CAREFULLY BEFORE BUYING!!!

The difference between a good programmer and an awesome programmer:

- **A good Programmer: works really hard**
- **An awesome programmer: Upsolves**

List of resources:

- Blog by Triveni Mahatha
- Competitive Programmer's Handbook
- Main Page - Competitive Programming Algorithms
- Striver's SDE Sheet
- Quora
  - What is a list of data structures that a competitive programmer must know?
  - How do I learn competitive programming as a beginner?

- How can I become good at competitive programming?
- What is competitive programming?
- What are some good coding competition/practice sites?
- What is the best way to progress through practice problems on CodeChef, SPOJ, TopCoder, etc.?

- Codeforces
  - An awesome list for competitive programming!
  - StopStalk: Tool to maintain your algorithmic progress
  - If you ask me how to improve your algorithm competition skill, I will give you the link of this blog.
  - [Tutorial] A way to Practice Competitive Programming: From Rating 1000 to 2400+
  - Competitive Programmer's Handbook — a new book on competitive programming
  - The 'science' of training in competitive programming
  - YouTube Channels for Competitive Programmers
  - Radewoosh's blog
  - Topic-wise Coding Resources
  - CODEFORCES for BEGINNERS...
  - Competitive Programming Community Discord (5800+ Members): Discussions hub!
- CodeChef
  - Getting Started
  - Data Structures and Algorithms
  - Learn Data Structures and Algorithms
  - Useful Resources For Competitive Programming
  - Competitive programming best resource
  - The CodeChef Wiki
    - Tutorials
  - StopStalk: Tool to maintain your algorithmic progress
- Medium
  - How to prepare for competitive programming?
  - My Competitive Programming Journey To Google
- HackerEarth
  - The Complete Reference to Competitive Programming
  - Programming Tutorials and Practice Problems
  - HackerEarth trending notes
- GeeksForGeeks
  - How to begin with Competitive Programming?
  - Tips and Tricks for Competitive Programmers | Set 1 (For Beginners) - GeeksforGeeks
  - How to become a master in competitive programming?
- Topcoder
  - Competitive Programming Tutorials
- Coding Blocks Blog — Sites and Tools for Competitive Programming
- redgreencode - How do I get better at competitive programming?
- Shahjalal Shohag Blog - The Ultimate Topic List(with Tutorials, Problems, and Templates)
- Guru99 - Competitive Programming for Beginners
- blogspot - Algorithms Live!
- Online Courses
  - edX
    - How to Win Coding Competitions: Secrets of Champions
  - MIT OCW
    - Introduction to Algorithms
    - Design and Analysis of Algorithms
    - Advanced Algorithms
  - CMU 15-451 (Algorithms), Fall 2010
  - USC - Data Structures and Object-Oriented Design
  - Stanford - CS 97SI: Introduction to Programming Contests
  - Harvard - CS 224: Advanced Algorithms
- List of competitive programming sites
  - Codechef
  - Codeforces
  - Hackerrank
  - GeeksForGeeks
  - InterviewBit
  - PrepBytes
  - Hackerearth
  - AtCoder
  - Topcoder

        - UVa Online
        - Project Euler
        - Timus Online Judge
        - A2 Online Judge
        - Google's Coding Competitions Archives
        - ACM ICPC Problem Archives
        - Panda Online Judge
        - Zhejiang University Online Judge (ZOJ)
    - Coding Ninjas
    - Codecademy
    - My Code School
    - freeCodeCamp
    - Codility
    - Coderbyte
    - Codecup
    - CodeGround
    - CodeJam
    - Codewars
    - DoSelect for developers
    - GeekyPrep.com
    - Kattis
    - LeetCode
    - CodingBat
    - Programming Hub
    - TechGig
    - Techie Delight
    - The Python Challenge
    - Codingame: Coding Games and Programming Challenges to Code Better
    - HackerBlocks
    - AlgoExpert
    - PlacementSaga
- Tools for Programming Contests
    - Online IDE's
        - Ideone
        - LeetCode
        - Rextester
        - Codeforces
        - CodeChef
        - Coding Blocks
        - CSAcademy
        - HackerRank
- Notable Competitions
    - ACM-ICPC
    - IEEEXTREME
    - TCS CodeVita
    - Google Code Jam
    - Google Kick Start
    - Google Hash Code
    - CodeChef SnackDown
    - Microsoft Imagine Cup
    - Facebook Hacker Cup
    - Topcoder Open (TCO)
    - Yandex Algorithm
    - Hewlett Packard (HP) Codewars
- Books
    - CLRS
    - Competitive Programming 3
    - Competitive Programmer's Handbook
    - The Art of Computer Programming

If you made it till here, please feel free to suggest corrections or addition of resources.

- Tell us about a time when you overcame a rather difficult challenge pertaining to a past project.
- Tell us about a time when you had to halt a project halfway

... (more)

You upvoted this

Upvote · 1.8K 29 144

competitive programming?

Different people might have a different perception of starting from 0 to get very good at competitive programming. I respect all their opinions. Today, I would be sharing my point of view. You might **pick-up the points that you find helpful**.

Please follow this order sequentially

- **Follow the following topics on Quora:**
  - Competitive Programming
  - Algorithms
  - Computer Programming
- **Subscribe the following YouTube channels:**
  - Tushar Roy - Coding Made Simple
  - take U forward
  - WilliamFiset
  - Aditya Verma
  - Pepcoding
  - Code NCode
  - mycodeschool
  - Gaurav Sen
  - Rachit Jain
  - code_report
  - Abdul Bari
- **Study the style guide before starting to write the**

... (more)

Upvote · 1.5K 23 21

Related **How should I get started in competitive programming?**

Here's how I did:

1. Learn one programming language really well - preferably C++. Make sure you're super comfortable with all the constructs and can code up any algorithm given the pseudo code.
2. Go to Hackerrank. Easy problems for starters and a good UI. Get a badge or two in C++ and Problem Solving (5 stars in Hackerrank is not too tough to get).
3. Then go to SPOJ. Sort the problems in descending order of "number of people who have solved them". This is important as it will make sure the problem level increases gradually. After 20–30 questions the problems tend to become quite tough. Solve 100.
4. This sh

... (more)

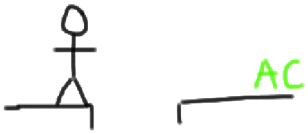
Aman Goel upvoted this

Upvote · 11.1K 44 128

in C++ in 2-3 months?

Many people tell you that solving lots of problems and you will become red on Topcoder/Codeforces one day. It is true, and is the only universally approved way in competitive programming community, but it is actually just half of the story. Let me first explain

For each problem, in order to solve it, you must jump over a gap. It can be either a difficult implementation, or some hard-to-see observation, difficult algorithm, etc.



For me, some problems are very easy (e.g. Codeforces div 2 ... (more)

Related questions More answers below

What should I know before starting competitive programming?

What are the prizes if you win a "competitive coding" competition?

How do I start competitive programming? What should I learn first, and what shall be my approach to be a good competitive programmer?

Upvote

Competitive programming is a challenging field that requires a combination of strong problem-solving skills, algorithmic thinking, and programming expertise. Here are some steps that can help you get started:

1. Learn a programming language: Choose a programming language that you are comfortable with and learn its syntax and basic concepts.
2. Practice coding: Practice coding regularly to improve your programming skills. There are many online platforms, such as HackerRank, Codeforces, and LeetCode, that offer coding challenges and problems to solve.
3. Learn algorithms and data structures: Competitive p

... (more)

Upvote · 5.4K 46 31

Related **What's your story as a competitive programmer?**

I started competitive programming in October 2013 with CodeChef's Long Challenge.

It was my third semester of engineering and I was having an easy time with my course curriculum. I didn't have any challenges to overcome and I didn't have anything to study. I spent my time playing a few easy computer games and watching TV Series full of violence.

Two very able seniors of mine Tushar Makkar [https://... (more)

You upvoted this

Upvote · 4.6K 52 80

If you Actually want start competitive coding then you must have to follow this method.

Step 1

Learn C++ or Java. If you can learn C, you can learn C++ and I recommend java first.

Why? Because C++ and Java both works on object oriented programming also these are not too complex languages.

Again,

Step 2

Get on Hackerrank because it has the best User Interface, combined with relative ease of getting started for the beginners.

Ease in the sense that anyone with zero CP/al
... (more)

Upvote · 557 4 6

Related **What are the benefits of learning competitive programming?**

Doing competitive programming (CP) at the expense of programming in core CS subjects like operating systems, networking, database management systems, computer architecture, compilers, distributed systems, object oriented programming, etc, is a disaster.

When you transition from student to employee, you don't get to solve single-threaded, fancy DS&A problems like in CP. Instead, as a fresher, you may find yourself scrubbing the UI or fixing some low priority bugs in the legacy backend. With limited systems background you are unlikely to make major feature contributions anyway.

Having said that, C... (more)

Upvote · 471

Upvote · 664 71 8

Related **Which one is the most basic programming language to learn, and why is that?**

- The most **basic** is **BASIC**. But it's not very useful these days, it's kind of obsolete.
- The most **useful** is **C**, it is very closely related to systems and thus forces you to learn computer systems.
- The most **straightforward** is **PHP**, although most tutorials out there are focused on Web programming requiring you to learn what Web is first!
- The most **complete** is **Python**. It has huge amount of libraries and vast uses. It also has a delightful specification and syntax.
- The most **used yet annoying** is **Javascript**. It's just.... Javascript.
- The most **popular** is **Java**. The relationship between Java and Javascript is like C

... (more)

Upvote · 875 29 14

Related **What made you good at competitive programming?**

I spent my college summer break at the hostel. I practiced a lot. Used to solve 30–45 problems a day. This included the solved sections of geeksforgeeks, hackerearth, hackerrank and many more. This also included coding from scratch for every solution to each problem. So, the number of programs I coded every day was above 45 for sure. I wasn't able to maintain this number after 12–15 days as I remember, because I already had solved easier ones and there were only a few left in that category. I don't remember how many days were there in that summer break but I spent min 10hrs a day (there was no... (more)

Upvote · 260 6 7

Related **What is a good schedule to follow for becoming better at competitive programming for beginners?**

The three most important things in competitive programming:

1. Learn and know your programming language. (~3 months - all the time)
2. Learn algorithms/data structures and implement. (~9 months - all the time)
3. Practice coding every day. Do contests. (all the time)
4. (Doesn't just apply to competitive programming) **Have fun!** (all the time)

Thus, that'll be how you break up your schedule. The time it takes for each person will differ, depending on if you already know algorithms, if you know programming or not, etc.

The following is a more detailed how:

#1 Learn and know your programming language.

Pick a programm... (more)

Upvote · 438 6 21

Related **What is your competitive programming journey (in short)?**

I am right now not that pro programmer so mine is not a journey to motivate others ,but i have some one else's journey to share.

ABDULLAH ASLAM.

He is neither from iit nor from nit even below tier-3 level college.

My Competitive Programming Journey To Google.

How it all started:
I've been interested in programming since class XI (I used to be the computer science topper XD), but it was limit to academics until my 2nd year in engineering. On a random day during my 3rd Semester, I received a call from Mehul (my classmate), who wanted to attend a team coding competition (at a college fest) but did not... (more)

Upvote · 112 4 17

Related **What are the benefits of learning competitive programming?**

The benefits of competitive programming are manifold. I am eligible to write this answer because I have been doing competitive programming from past 2 years and it did help me to a great extent.

Coming back to the question, what are the benefits of competitive programming?

1. If you have a good rating/ranks on websites like codeforces/codechef then you have a very high chance of getting an interview call from companies like google, facebook, Amazon, Directi etc. It also depends a lot on your resume though.
2. You get to participate in competitions like ACM-ICPC and google code jam and get to meet very

... (more)

Aman Goel upvoted this

Upvote · 958 12 5

- Competitive programming problems are usually precisely specified. Real world applications usually are not. The programmer (or someone else) has to make design decisions.
- Competitive programming problems have strict constraints (can only use 512 MB of RAM, must run in under 2 seconds, and so on). Real world applications usually do not. The programmer has to make tradeoffs, often after consulting with a product manager.
- Competitive programming problems almost always have known solutions. Someone else knows the solution, and the question is whether you are smart enough to also find it. In real w

... (more)

1 answer collapsed (Why?)

Related questions

How much time do I need to be a competitive programmer?

What is the way to get better at competitive programming in 1 month with some (little) previous experience?

What are the best websites that let you learn and practice coding simultaneously?

What's your story as a competitive programmer?

What is the best way to begin my competitive programming experience?

What are some tips for competitive programming?

ago, also if you have any other advice if would be really helpful.

What is the dark side of competitive programming?

What is a good way to order topics in terms of what to learn first for competitive programming?

How do I get good at competitive programming quickly?

How much time does it take to learn C++ if you already have good knowledge in C?

How has competitive programming helped you get a job?

What are the steps for getting started in competitive programming, which sites should I use first, and what strategy should I follow to become good in competitive programming?

What is the best way to get good at competitive programming?

How much of C++ should I learn for competitive programming?