



Introduction to Competitive Programming

Today's Agenda

1. What is CP?
2. Its benefits
3. Pre requisites
4. How to increase rating
5. Live Coding

Competitive Coding



- Algorithmic problem solving
- Analytical thinking
- Implementing what you think

Competitive Coding



- Algorithmic problem solving
- Analytical thinking
- Implementing what you think

Competitive Coding



- Algorithmic problem solving
- Analytical thinking
- Implementing what you think

But why?





But why?

- Problem-solving skills improvement
- Placement/Internship
- Online Coding Round
- Easy referrals



But why?

- Problem-solving skills improvement
- Placement/Internship
- Online Coding Round
- Easy referrals



But why?

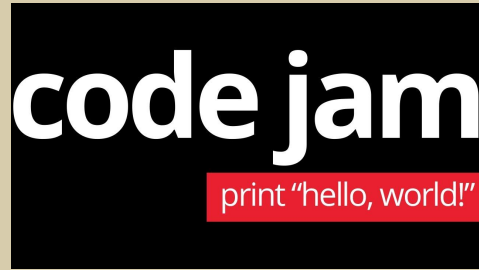
- Problem-solving skills improvement
- Placement/Internship
- Online Coding Round
- Easy referrals



But why?

- Problem-solving skills improvement
- Placement/Internship
- Online Coding Round
- Easy referrals

Popular contests



...etc



Pre-Requisites

- Language
(C++/Java/Python)
- Basic DSA

Platforms



...etc

Contests on



Long Challenge

- 10 days
- First Friday of every month

Cook-off

- 2.5 hrs
- Second last Sunday of every month

Lunchtime

- 3 hrs
- Last Saturday of every month

Contests on



Long Challenge

- 10 days
- First Friday of every month

Cook-off

- 2.5 hrs
- Second last Sunday of every month

Lunchtime

- 3 hrs
- Last Saturday of every month

Contests on



Long Challenge

- 10 days
- First Friday of every month



Cook-off

- 2.5 hrs
- Second last Sunday of every month

Lunchtime

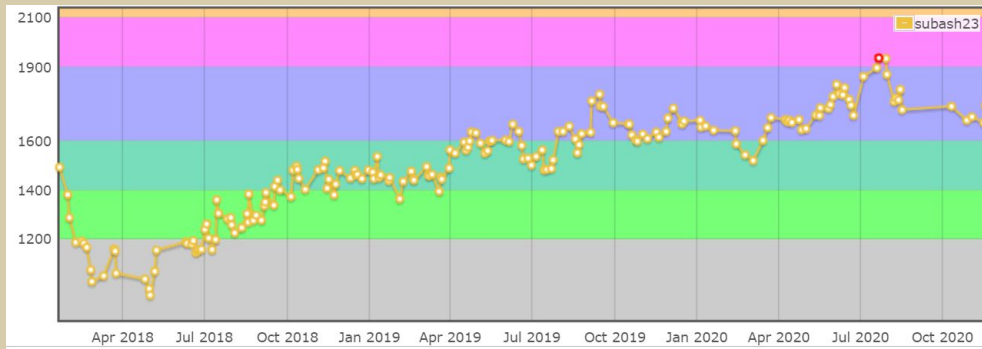
- 3 hrs
- Last Saturday of every month

Where to practice questions from?

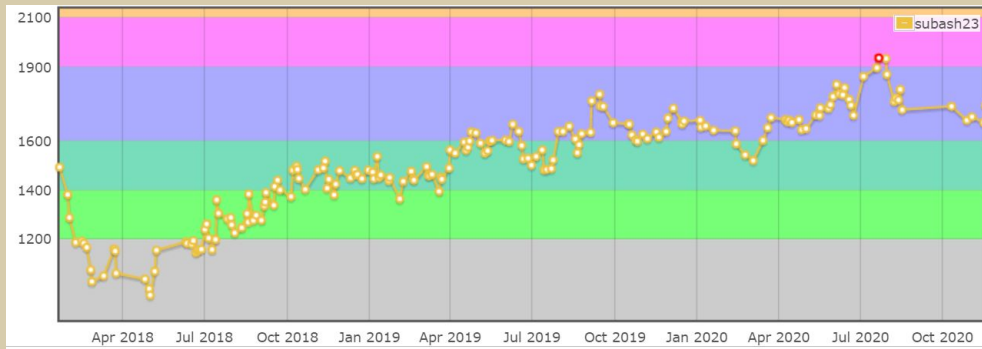
- Search by tags on  and  **CODEFORCES**
- a2oj.com/ladders
- Atcoder Beginner Contests
- algoshots.herokuapp.com
- projecteuler.net
- www.geeksforgeeks.org
- cp-algorithms.com
- Competitive Programmer's Handbook by Antti Laaksonen
Related Problems: cses.fi/problemset

How to increase rating?

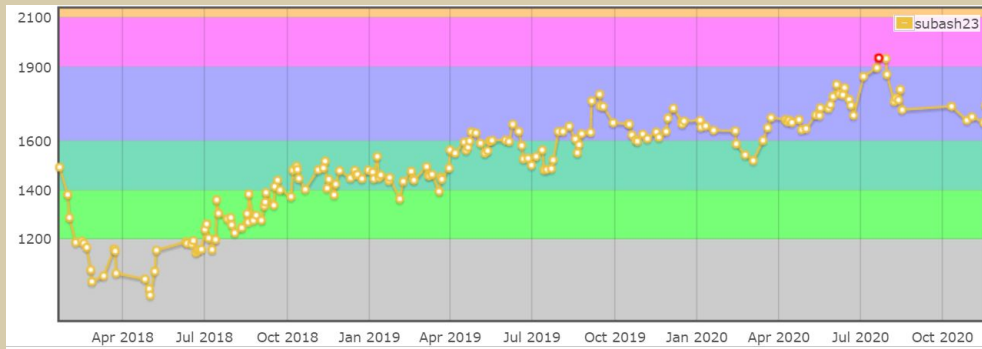




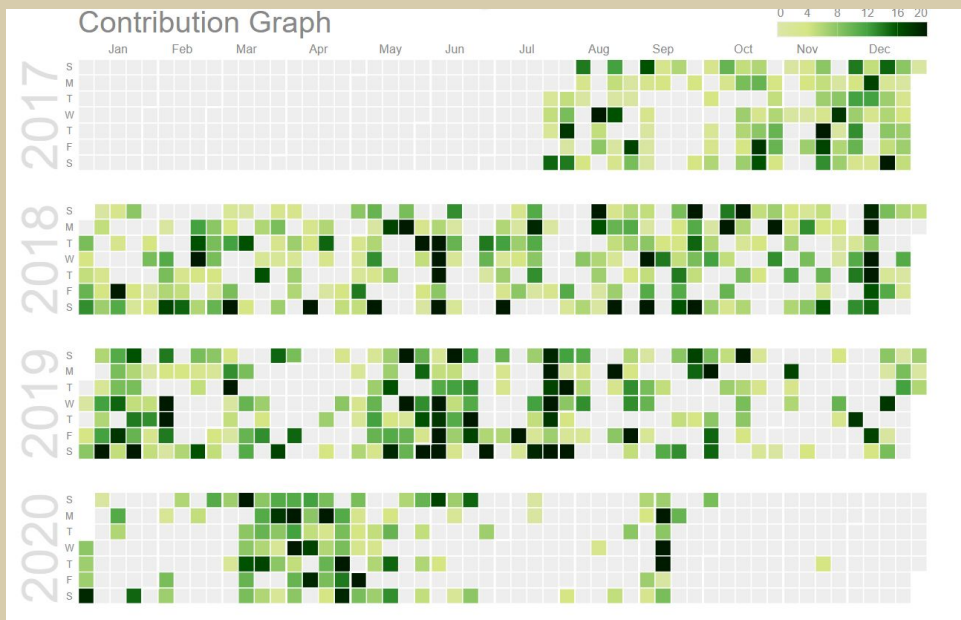
- Participate in lot (every) of contests
- Upsolve after each contest and learn new concepts from them
- Practice questions from practice section
- Consistency is the key



- Participate in lot (every) of contests
- Upsolve after each contest and learn new concepts from them
- Practice questions from practice section
- Consistency is the key



- Participate in lot (every) of contests
- Upsolve after each contest and learn new concepts from them
- Practice questions from practice section
- Consistency is the key



- Participate in lot (every) of contests
- Upsolve after each contest and learn new concepts from them
- Practice questions from practice section
- Consistency is the key

Final Tips

- Read the problem statement quickly
- Have a pen and paper handy
- Use code snippets
- Dry run program before submitting
- Check for corner cases
- Don't focus much on ratings
- Peer programming



cutt.ly/introtocp

Let's code 🦾

