OMMC

Rutgers Exposition in Problem Solving

Christian Lim

Saturday, June 7, 2024

Christian Yongwhan Lim

Education

Stanford University

MIT

Part-time Jobs

Cornell Tech

MIT EECS

Full-time Job

Google Research

TWO SIGMA

Workshops

Stanford ENGINEERING | Stanford Computer Forum

Carnegie Mellon University

Harvard

University of California

KAIST

NUS National University of Singapore

Coach/Judge

icpc International Collegiate Programming Contest

icpc.foundation

advancing the art and sport of competitive programming

https://www.yongwhan.io

Christian Yongwhan Lim

- **Vice President (VP) of Platform**, Arklex AI

- **Adjunct Associate Professor of Computer Science**, Columbia University
- **Coach**, Columbia International Collegiate Programming Contest (ICPC) Team

- **Director**, ICPC Internships, ICPC Foundation

https://www.yongwhan.io

As you know, topics in typical math competition are:

- Ad hoc
- Logic
- Algebra
- Geometry
- Combinatorics
- Number Theory
- Sequences and Series
- Graph Theory
- Probability
- ...

Those topics are ALSO in programming competitions!

- Ad hoc
- Logic
- Algebra
- Geometry
- Combinatorics
- Number Theory
- Sequences and Series
- Graph Theory
- Probability
- ...

Except... you just need to implement your solution!

- Biggest Advantage: **checking if your solution is correct is automated**!
  - **MUCH** easier to check your understanding!!

Except... you just need to implement your solution!

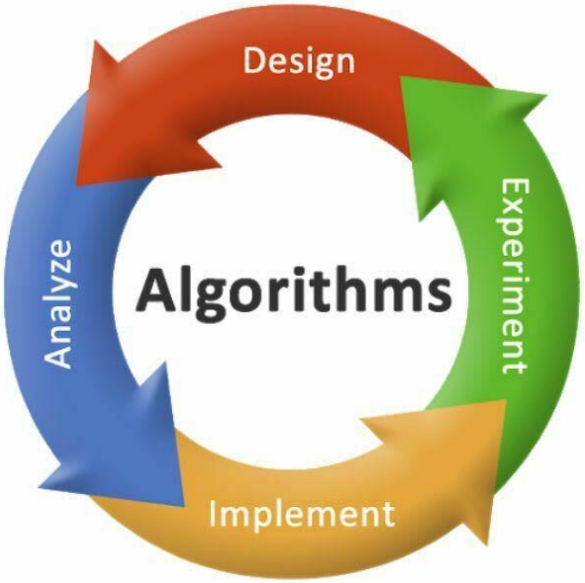
- Biggest Advantage: **checking if your solution is correct is automated**!
  - **MUCH** easier to check your understanding!!

- Also, it is often **more concrete** than typical mathematical proofs, which may often be (quite) abstract!
  - Some may see this as a **plus**; some as a **minus**.
  - I used to think this is a minus; but, my perspective changed completely over the years!

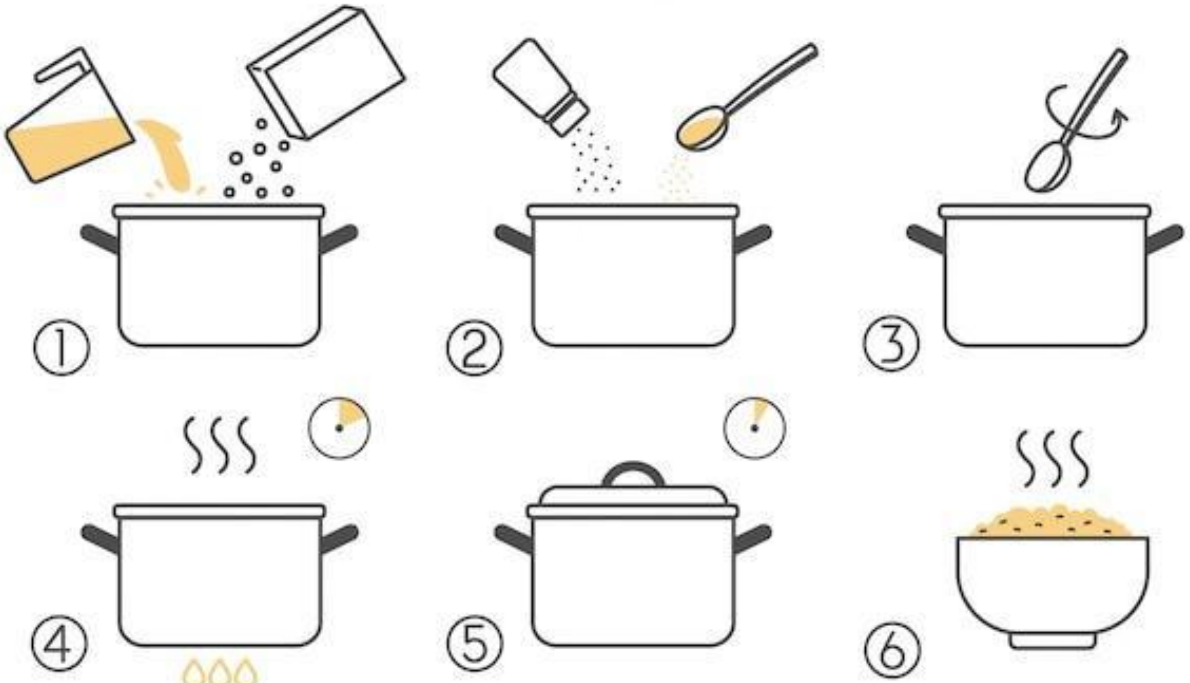
So, what exactly is an algorithm?

So, what exactly is an algorithm?

- A set of **step-by-step procedures**, or a set of rules to follow, for completing a specific task or solving a particular problem.



HOW TO COOK PORRIDGE



Why do we study algorithm?

- **To solve standard problems efficiently!**
  - Typically, efficiency can mean the program runs faster or uses less memory or both!

Why do we study algorithm?

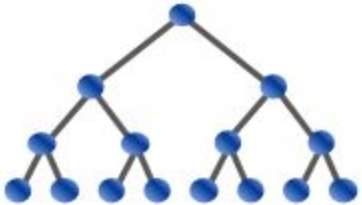
- **To solve standard problems efficiently!**
  - Typically, efficiency can mean the program runs faster or uses less memory or both!
- **To become a better programmer!**
  - This can help you win programming (or, also, math) contests!

Why do we study algorithm?

- **To solve standard problems efficiently!**
  - Typically, efficiency can mean the program runs faster or uses less memory or both!
- **To become a better programmer!**
  - This can help you win programming (or, also, math) contests!
- **But, most importantly, TO HAVE FUN!**
  - Solving problems can be fun!

USA Computing Olympiad (USACO)

USA Computing Olympiad



If selected, International Olympiad in Informatics (IOI)

Meta Hacker Cup

International Collegiate Programming Contest (ICPC)

Popular Contest Sites

AtCoder

CODEFORCES β
Sponsored by Telegram

topcoder ™

Popular Practice Sites

Popular Tutorial Sites

usaco.guide

cp-algorithms.com

Programming Zealots @Discord

- Break into **CodeForces** rating of **2200+** as fast as you can!

- Join the discord server!

  https://bit.ly/programming-zealot

Success Pathways

- **Programming Zealots @ CodeForces**
- **800 - 2100 (A - N)**
  - **For those who are just starting**
  - To gain some experiences with an explicit goal to enjoy the process of solving new problems;
  - To make it to bronze, silver, gold, and platinum in USACO!

Success Pathways

- **Programming Zealots @ CodeForces**
- **800 - 2100 (A - N)**
  - **For those who are just starting**
  - To gain some experiences with an explicit goal to enjoy the process of solving new problems;
  - To make it to bronze, silver, gold, and platinum in USACO!
- **2200 - 3500 (O - ZB)**
  - **For those who are more serious**
  - To make it to USACO training camp or IOI!

Practice Strategy

- If your goal is to get to a rating of **X**, you should practice on problems that are **X + 300** typically, with a spread of 100. So, picking problems within the range of:

  {X + 200, X + 300, X + 400}

  would be sensible!

Practice Strategy

- If your goal is to get to a rating of **X**, you should practice on problems that are **X + 300** typically, with a spread of 100. So, picking problems within the range of:

  {X + 200, X + 300, X + 400}

  would be sensible!

- So, if you want to target becoming a **red (grandmaster)**, which has a lower-bound of 2400, you should aim to solving {2600, 2700, 2800}.
- **(Eventual) Target**: You should focus on solving it for 30 minutes or less!

Practice Strategy (con't)

- You should focus on solving each problem for **30 minutes or less**; if you cannot, you should consider solving a problem with a lower rating.
- You should aim to solve **~5 problems** each day within this range to expect a rank up within six months.

Practice Strategy (con't)

- You should focus on solving each problem for **30 minutes or less**; if you cannot, you should consider solving a problem with a lower rating.
- You should aim to solve **~5 problems** each day within this range to expect a rank up within six months.
- If you cannot solve a problem, here is a sample recipe you can follow:
  - Look at editorial for **hints**, and try to solve the problem.
  - Look at editorial for **full solutions**, and try to solve the problem.
  - Look at **accepted code**, and try to solve the problem.
  - Make sure you **revisit after two weeks** and see if you can solve it.

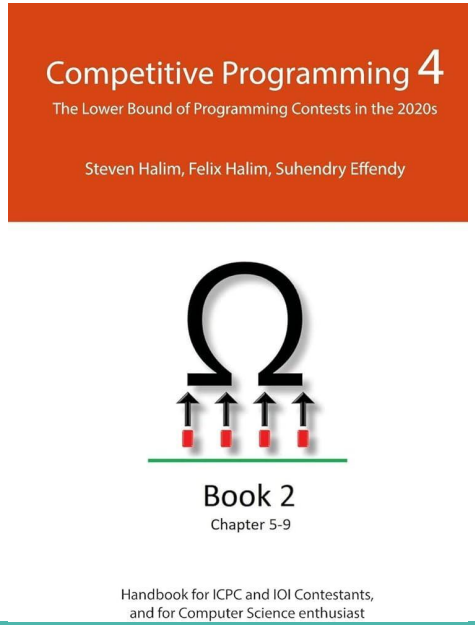
Training Resources

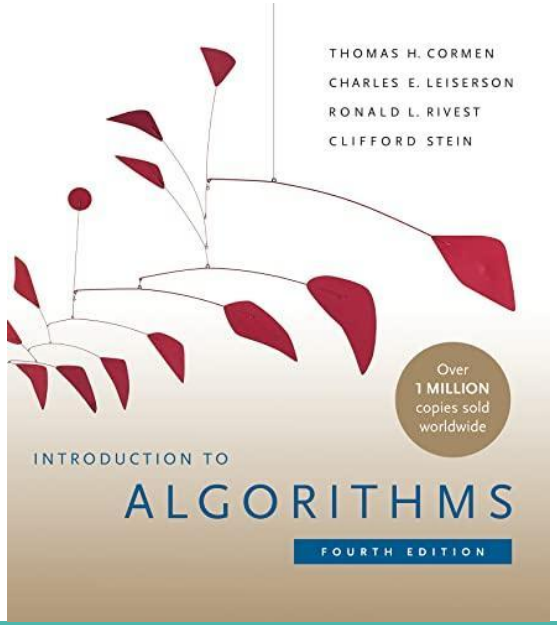
- **U ICPC**: https://u.icpc.global/training/
- **CP Algorithms**: https://cp-algorithms.com/
- **USACO Guide**: https://usaco.guide/

- **Kattis**: https://open.kattis.com/
- **Methods to Solve**: https://cpbook.net/methodstosolve?oj=kattis&topic=all&quality=all
- **CSES**: https://cses.fi/problemset/

Textbooks

- **Competitive Programming 4**, Halim, et. al.
- **Introduction to Algorithms**, Cormen, et. al.





Growing Short List of Useful Websites

- u.icpc.global/training

Programming Language Choice

- For now, probably **one** of the following languages:
  - **C++**
  - Java
  - Python

Programming Language Choice

- For now, probably **one** of the following languages:
  - **C++**
  - Java
  - Python

- It is the best to pick **C++** if you would like to be a serious (competitive) programmer.

You may find the slide deck today from my GitHub

- https://github.com/yongwhan/yongwhan.github.io

1:1 Quick Chat

- You may use https://calendly.com/yongwhan/one-on-one to sign up!

Ask me anything!

- **Email**: yongwhan@yongwhan.io

- **(Personal) Website**: https://www.yongwhan.io/

- **LinkedIn Profile**: https://www.linkedin.com/in/yongwhan/
  - Feel free to send me a connection request!
  - Always happy to make connections with awesome students! :)