OMMC

Rutgers Exposition in Problem Solving

Christian Lim

Sunday, April 7, 2024

Christian Yongwhan Lim

Education

Stanford University

Part-time Jobs

Full-time Job

Google Research

TWO SIGMA

Workshops

Stanford ENGINEERING | Stanford Computer Forum

Coach/Judge

icpc International Collegiate Programming Contest

icpc.foundation

advancing the art and sport of competitive programming

https://www.yongwhan.io

Christian Yongwhan Lim

- Christian and Grace Consulting **Owner**;
- ICPC **Internship Manager**;
- ICPC North America **Leadership** Team;
- ICPC North America Championship **Operations**;
- ICPC North America Programming Camp **Trainer**;
- ICPC NAQ and Regionals **Judge**;
- ICPC World Finals CLI Symposium **Co-lead**;
- ICPC Curriculum Committee **Co-lead**;
- Columbia ICPC **Head Coach**;
- Columbia **Adjunct** (Associate in CS);

https://www.yongwhan.io

Road to International Competitions

The Full Contest Math Pathway

Acronym Key

MOEMS: Mathematics of Olympiads for Elementary and Middle Schools

USAMTS: USA Mathematical Talent Search

AMC: American Mathematics Contest

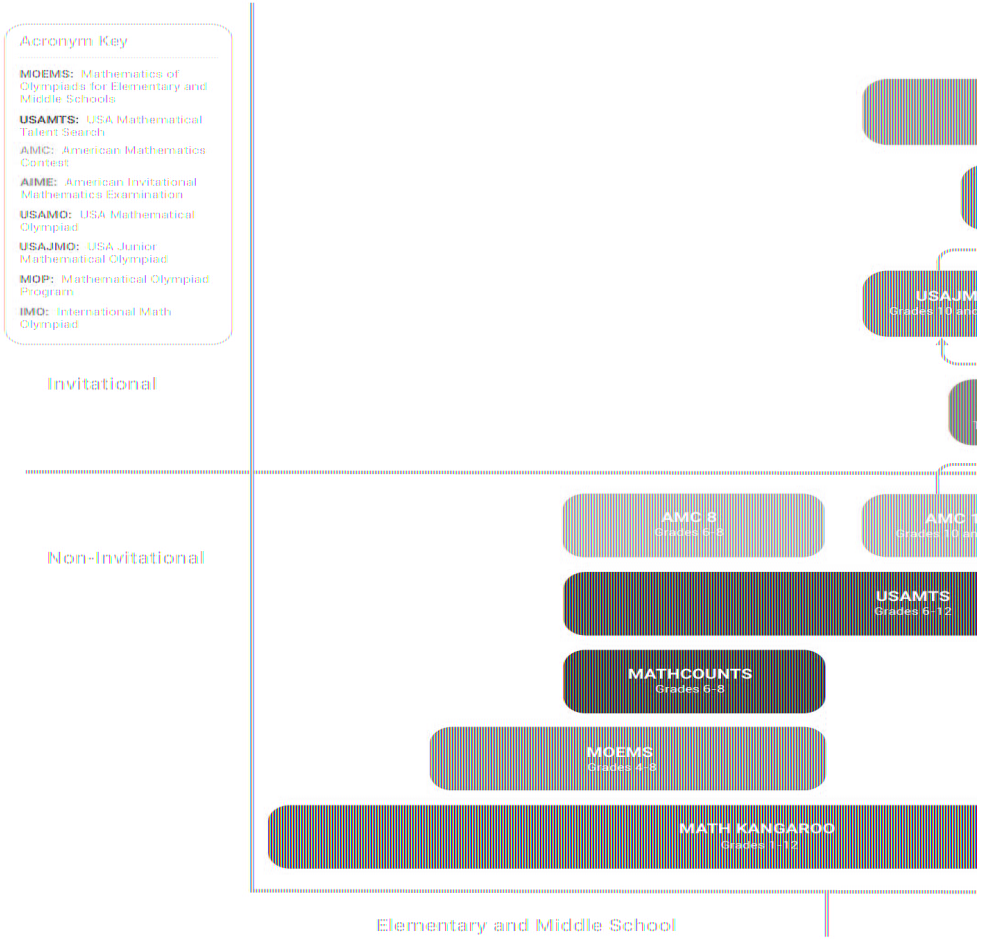
AIME: American Invitational Mathematics Examination

USAMO: USA Mathematical Olympiad

USAJMO: USA Junior Mathematical Olympiad

MOP: Mathematical Olympiad Program

IMO: International Math Olympiad



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!

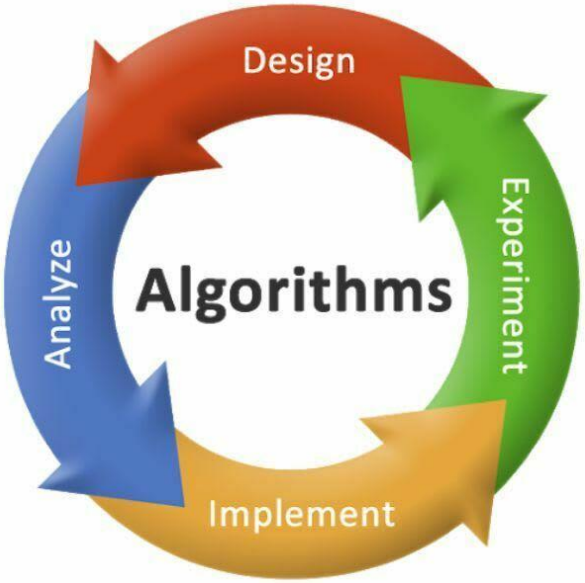
- Biggest Advantage: **checking if your solution is correct is automated**!
  - **MUCH** easier to check your understanding!!

- 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!

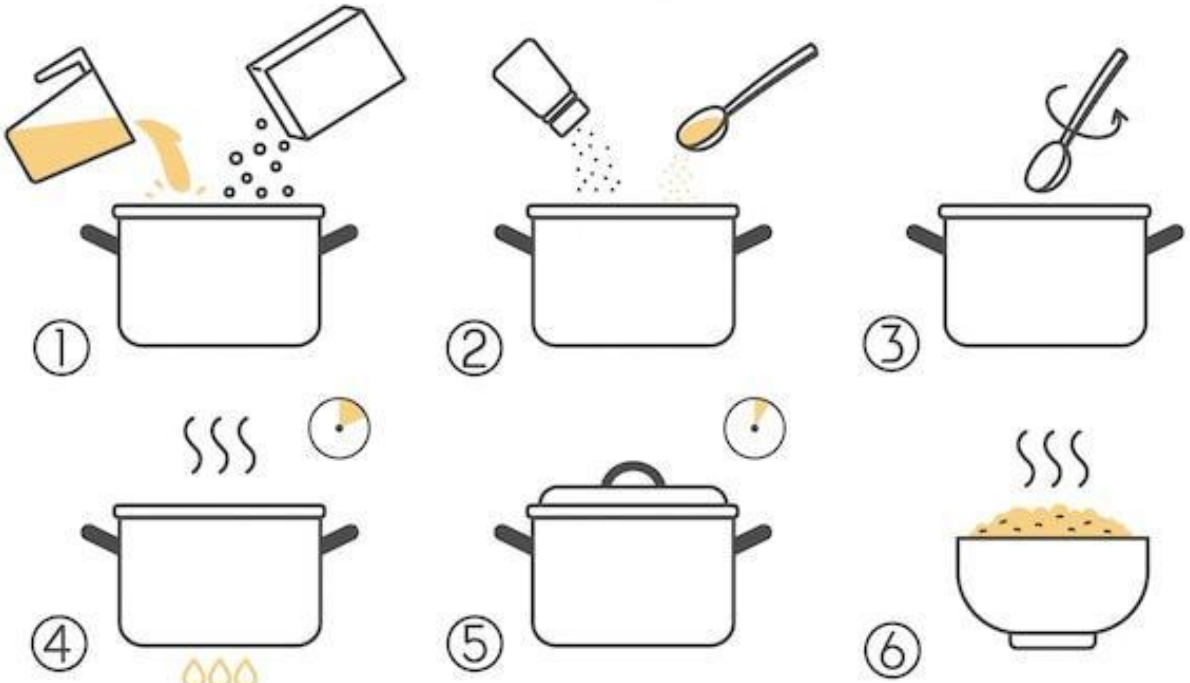
So, what exactly is an algorithm?

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?

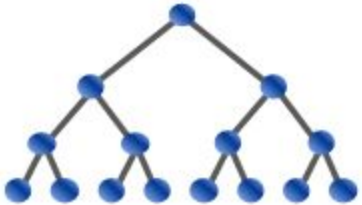
- **To solve standard problems efficiently!**
  - Typically, efficiency can mean the program runs faster or uses less memory or both!
- **To become a better programmer!**
  - This can help you win programming (or, also, math) contests!
  - This can help you with an early start on an interview preparation!

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!
  - This can help you with an early start on an interview preparation!
- **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

Programming Zealots @CodeForces

- Also, join CodeForces group!

bit.ly/cf-zealots

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.

More on Programming Contests!

- **Universal Cup**: **https://ucup.ac/register**

- **Quarterly Contests** from ICPC Curriculum Committee, starting **June 2024**

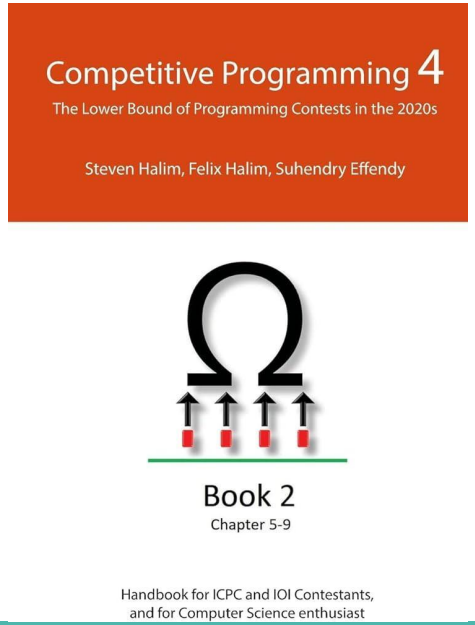
Training Resources

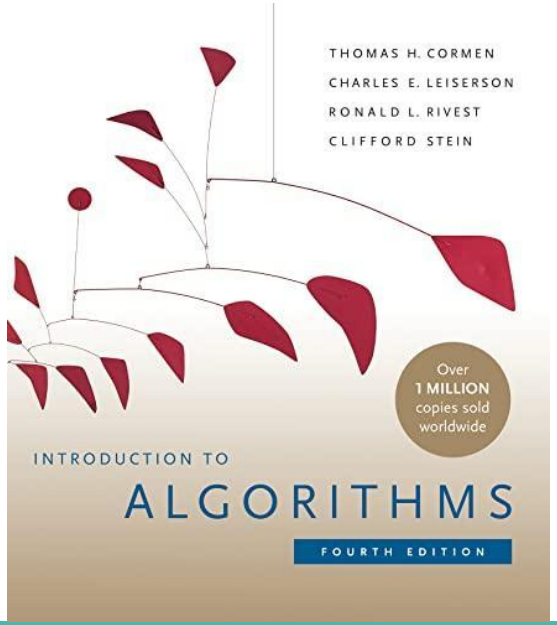
- **U ICPC**: https://u.icpc.global/training/
- **CP Algorithms**: https://cp-algorithms.com/
- **USACO Guide**: https://usaco.guide/

- **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.





More on Growing Short List of Useful Websites

- Please take a look as needed: **Link**
- Alternatively, you can also get to this from **u.icpc.global/training**!
- You may also take a look at terse guides I have written over the years!

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.

Opportunities @Columbia

- **SHP** (Science Honors Program)
- **SHAPE** (Summer High School Academic Program for Engineers)

- **SHSCPC** (Summer High School Coaching for Programming Contests)

- If you would like to know more about them, please send an email to yongwhan.lim@columbia.edu!

Columbia University Local Contest (CULC)

- 3rd **Columbia University Local Contest** (CULC)
  - **Individual**, not team, contest!
  - Date: **Saturday, April 27, 2024**
  - Time: **1pm ET ~ 6pm ET**
  - @**Uris Hall, Columbia University**

- **https://bit.ly/spring2024-culc-flyer**

1:1 Quick Chat

- You may use https://calendly.com/yongwhan/quick-chat-blitz to sign up!

Questions and Answers!

- **Ask me anything!**

You may find the slide deck today from

- **Christian Lim**'s github page

- **Direct Link**
  - https://github.com/yongwhan/yongwhan.github.io/tree/master/omm c-rep

Contact Information

- **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! :)