# Pacific Northwest ICPC
# Problem Solving Workshop

Saturday, February 10, 2024

**Christian Lim**

# 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

- Currently:
  - **Internship Manager**, ICPC Foundation;
  - **Leadership Team**, ICPC North America (NA);
  - **Trainer**, ICPC NA Programming Camp;
  - **Judge**, ICPC NA Qualifiers and Regionals;
  - **Adjunct**, Columbia CS;
  - **CEO** (Co-Founder), Stealth Mode Startup;
  - **Co-Founder**, Christian and Grace Consulting;
  - **Head Coach**, Columbia ICPC;

https://www.yongwhan.io

# Introduce who YOU ARE!

- What year are you?
- What major?
- What university?
- What is your hobby?
- What is your plan after graduation?

# Overview

- Part I: **General Introduction**
- Part II: **Interview Preparation Guide**

# Popular Contest Sites

AtCoder

CODEFORCES β
Sponsored by Telegram

topcoder ™

# Popular Practice Sites


# Popular Tutorial Sites


usaco.guide

cp-algorithms.com

# More on Growing Short List of Useful Websites

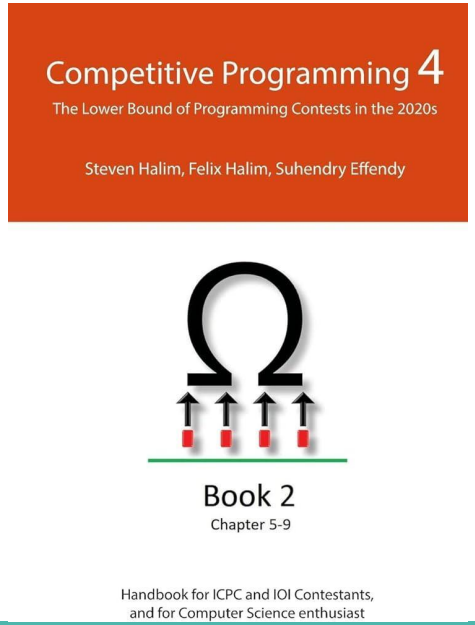
- Please take a look as needed: **Link**

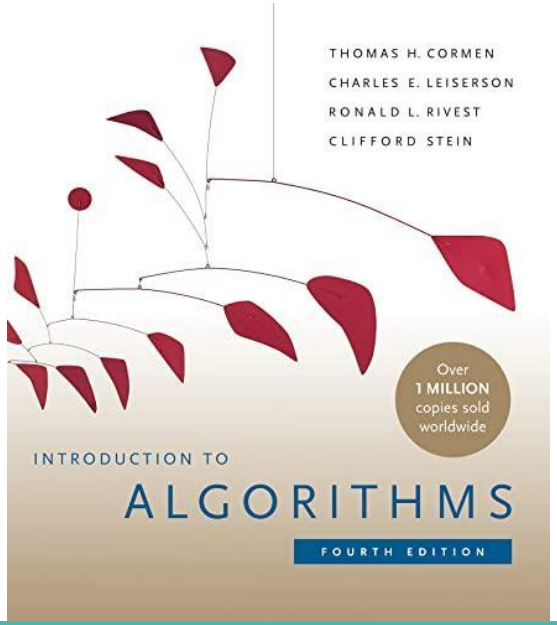
# Terse Guides

- Please take a look as needed: **Link**

# Textbooks

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



# Programming Language Choice

- You are welcome to pick 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.

# 1:1 Quick Chat

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

# Discord Invitation => CodeForces Group

- Please join Programming Zealots using **https://discord.gg/Xea8BeHczd**!

- I will invite you to CodeForces group where you will get weekly/weekend problem sets!
- In addition, you may join the **masterclasses** on **Sundays** to learn more about how to solve the problems from the weekend problem set!

# Practice Strategies in CodeForces

- If your goal is to get to a CodeForces 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**, 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 Strategies in CodeForces

- You should focus on solving each problem for **30 minutes or less**; if you cannot solve any problem with this range, you should consider solving a problem with a lower rating.

- You should aim to solve **10 ~ 15 problems** each day within this range to expect a rank up within a quarter (3 months).

# Practice Strategies

- 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.

# Practice Strategies

- 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.

# Practice Strategies

- 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 solutions, and try to solve the problem.

# Practice Strategies

- 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 solutions, and try to solve the problem.
  - Make sure you look back after two weeks and see if you can solve it.

# Live Contest Strategies

- A Terse Guide to Live Contests

# C++ Tips and Tricks: best to learn those through practice!

- **C++ Tricks (HosseinYousefi)**
- **C++ tips and tricks (Golovanov399)**
- **Some Tips for Coding in C++ in Competitive Programming (Nea1)**

- Use "**#include <bits/stdc++.h>**" header to include **almost everything**.


BREAK

# Interview Preparation Guide

# Interview Types

- **Technical Interview**
  - Tests technical skill-sets required for a job.
  - **Algorithms and Data Structures** (A & DS)
  - **System Designs**

- **Behavioral Interview**
  - Tests soft skills (e.g., effective communication, conflict resolution, etc)

# Technical Interview

- Recruiter Call
- 0-1 Online Coding Challenge
  - automated screening with 2-3 questions.
- 2-3 Technical Phone Screens
  - first technical conversation with human.
- 4-7 Interviews in Onsite
  - similar to phone screening but more in-depth; you may get probed on your claimed expertise.
- 0-5 Fit Calls & Negotiation

# Technical Interview

- Recruiter Call
- **0-1 Online Coding Challenge**
  - automated screening with 2-3 questions.
- **2-3 Technical Phone Screens**
  - first technical conversation with human.
- **4-7 Interviews in Onsite**
  - similar to phone screening but more in-depth; you may get probed on your claimed expertise.
- 0-5 Fit Calls & Negotiation

# I. ADS: Interview Topics

- **Fundamentals**
  - Arrays and Linked Lists
  - Binary Trees
  - Heaps
  - Sorting

# I. ADS: Interview Topics

- **Important**
  - Stacks and Queues
  - Hash Tables
  - Binary Search Trees
  - Searching
  - Recursion

# I. ADS: Interview Topics

- **Real Differentiators (Tech vs Quant)**
  - **Strings**: Knuth Morris Pratt (KMP); Rabin Karp / String Hashing; Suffix Array; Suffix Automaton;
  - **Dynamic Programming**: 1D; 2D; Interval; Tree;
  - **Greedy Algorithms** and Invariants: Matroid;
  - **Graphs**: Shortest Path; Lowest Common Ancestor; Flow / Matching; Minimum Spanning Tree;

# I. ADS: Interview Topics

- **Real Differentiators (Tech vs Quant)**
  - **Strings**: Knuth Morris Pratt (KMP); Rabin Karp / String Hashing; Suffix Array; Suffix Automaton;
  - **Dynamic Programming**: 1D; 2D; Interval; Tree;
  - **Greedy Algorithms** and Invariants: Matroid;
  - **Graphs**: Shortest Path; Lowest Common Ancestor; Flow / Matching; Minimum Spanning Tree;
    - BFS; DFS; Dijkstra; Bellman-Ford; Floyd-Warshall;
    - Ford-Fulkerson/Edmond-Karp; Dinic;
    - Prim; Kruskal (DSU);

# Warm-up Problem on String

- A minimum number of insertions to make a string a palindrome.

# Warm-up Problem on String

- A minimum number of insertions to make a string a palindrome.

- **Constraint**
  - string length is at most 5000
  - each character is from ‘a’ to ‘z’

# Warm-up Problem on String

- A minimum number of insertions to make a string a palindrome.

- **Constraint**
  - string length is at most 5000
  - each character is from ‘a’ to ‘z’

**Any Idea?**

# Model Solution

```
int minInsertions(string &s) {
    int n = s.size();
    vector<vector<int>> dp(n, vector<int>(n,0));
    for (int i = 1; i < n; i++)
        for (int j = 0, k = i; k < n; j++, k++)
            dp[j][k] = (s[j]==s[k]) ?
                            dp[j+1][k-1] :
                            min(dp[j][k-1],dp[j+1][k])+1;
    return dp[0][n-1];
}
```

# Interview Preparation Resources (Tech)

- **Popular Websites**
  - LeetCode: Solve all four weekly/biweekly problems in **60 minutes**!
    - 3+6+12+24 (+15 buffer)
  - CodeForces: Get to 1800+ rating
    - Clear 4 questions out of 6!
  - AtCoder; TopCoder; CodeChef;

- **Annual Contests**
  - Meta Hacker Cup; ~~Google Code Jam; TopCoder Open;~~

# Interview Preparation Resources (Quant)

- **Popular Websites**
  - LeetCode: Solve all four weekly/biweekly problems in **20 minutes**!
    - 1+2+4+8 (+5 buffer)
  - CodeForces: Get to 2200+ rating
    - Clear 5 questions out of 6 **fast**!
  - AtCoder; TopCoder; CodeChef;

- **Annual Contests**
  - Meta Hacker Cup; ~~Google Code Jam; TopCoder Open;~~

# Interview Preparation Resources

- Elements of Programming Interview
- Competitive Programming 4

# II. Behavioral Interview (for everyone)

- Becoming an industry standard to have at least one session in typical software engineering interview loop.
- Wants to assess leadership potential.
- Tests soft skills (e.g., effective communication, conflict resolution, etc.)
- Open-ended: **not** about getting it right or wrong!

# Example Question #1

- Tell me about a time when you led a team to successfully complete a project.

# Example Question #1: Sample Answer

- Best if you led a hackathon/passion project.
- Otherwise, if you led a project as an intern, highlight it.

- Be **concise**!
- Include hard **metrics** in terms of %, $, etc.
- Provide **concrete** examples.

# Example Question #2

- How do you set up priorities for the work you are facing each day?

# Example Question #2: Sample Answer

- Priority queue idea:
    - Most essential responsibilities first!
    - Respond to emergencies as needed.
    - Non-essential tasks can be delayed.

# Example Question #3

- What experiences do you have relevant to this job?

# Example Question #3: Sample Answer

- Highlight a technical project you have done that lasted **at least** one year.

- Discussing technologies is a **must**!
  - **Programming languages**: C++ vs Java vs Python vs Go vs ?
  - **Databases**: SQL vs NoSQL vs ?
  - **Algorithms and Data Structures**
  - **Development tools**: Emacs vs Vim vs Visual Studio vs JetBrain vs ?

# Resources

- There are number of preparation books.

- For example:
  - *Behavioral Interview Questions and Answers* by Horatio Bird;
  - *Leadership Interview Questions You'll Likely Be Asked* by Vibrant Publishers;

# III. System Design Interview (for > entry level)

- Identify large components of the system and describe how each component is connected.

- Actual implementation details are **not** as important.

- Tests whether you can design an architecture using standard design patterns.

# Resources

- Must reads are:
    - The System Design Interview, 2nd edition by Lewis C. Lin, et. al.
    - System Design Interview by Alex Xu

# @YOUR University?

- If you would like me to present this material (and more!) at your university, please send me a quick chat request through https://calendly.com/yongwhan/quick-chat.

# 1:1 Meeting Opportunity

- If you would like to meet in 1:1, please sign up using: **https://calendly.com/yongwhan/quick-chat**.

- **I'd love to help you landing your dream job!**

# Internship @ICPC Foundation

- If you would like to get involved in helping out as a volunteer or an official (unpaid) intern, please reach out to me with your resume at **internship@icpc.foundation**.

# Questions and Answers

- Ask me anything!

# 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 promising students!


THANK YOU