

---

---

# **Special Workshops** **for Technical Interviews**

— Yongwhan Lim —  
11am CET, Monday, June 12, 2023

---

---

# Yongwhan Lim



## Education



## Part-time Jobs



## Full-time Job



## Workshops



## Coach/Judge



<https://www.yongwhan.io>

# Yongwhan Lim



- Currently:
  - **CEO** (Co-Founder) in a Stealth Mode Startup;
  - **Co-Founder** in Christian and Grace Consulting;
  - **ICPC Internship Manager**;
  - **ICPC North America Leadership Team**;
  - **Columbia ICPC Head Coach**;
  - **ICPC Judge** for NAQ and Regionals;
  - **Adjunct** (Associate in CS) at Columbia;



<https://www.yongwhan.io>

# Overview

- **Part I: Interview Preparation**
  - Interview Types
  - Technical Interview
  - Interview Topics
  - Interview Preparation Resources
- **Part II: Competitive Programming**
  - CodeForces
  - ICPC

# Part I: Interview Preparation

# Interview Types

- Technical Interview
  - Tests technical skill-sets required for a job.
- Behavioral Interview
  - Tests soft skills (e.g., effective communication, conflict resolution, etc)

# Interview Types

- **Technical Interview**
  - Tests technical skill-sets required for a job.
- **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

# Interview Topics

- Data Structures and Algorithms
- (> entry level) System Design Problems

# Interview Topics

- **Data Structures and Algorithms**
- (> entry level) System Design Problems

# Interview Topics

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

# Interview Topics

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

# Interview Topics

- **Real Differentiators**

- **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; Flow / Matching; Minimum Spanning Tree;

# Interview Topics

- **Real Differentiators**

- **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; Flow / Matching; Minimum Spanning Tree;
  - SP: Floyd-Warshall; Dijkstra; BFS/DFS; Bellman-Ford
  - Flow: Dinic / Edmond-Karp + Ford-Fulkerson; MCMF;
  - MST: Kruskal (Disjoint-Set Union); Prim;

# Interview Preparation Resources

- **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**

# Part II: Competitive Programming

# CodeForces

- Get to **2200+** rating as fast as you can!

# 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**, which has a lower-bound of 2400, you should aim to solving  $\{2600, 2700, 2800\}$ .

# Practice Strategy

- 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 Strategy

- 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.

# Selected Links

- <https://cp-algorithms.com/>
- <https://usaco.guide/>
- [Terse Guides](#)
  - [\(Growing\) A Terse List of Useful Resources](#)

# 1:1 Mentoring Opportunity

- If you would like to meet in 1:1, please sign up using:  
<https://calendly.com/yongwhan/mentoring-fit>.
- I'd love to help you landing your dream job!



# Contact Information

- Email: [yongwhan@yongwhan.io](mailto: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!

# Slide Decks

- You can find the slide deck from this presentation here soon:

<https://github.com/yongwhan/yongwhan.github.io/tree/master/kth>