
Stony Brook University

Technical Interview Workshop

Part I

— **Christian Lim** —

1pm ET, April 2, 2024

Christian Yongwhan Lim



Education



Part-time Jobs



Full-time Job



Workshops



Coach/Judge



<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 North America Curriculum Committee **Co-lead**;
- Columbia ICPC **Head Coach**;
- **Adjunct** (Associate in CS) at Columbia;



<https://www.yongwhan.io>

Overview

- **Part I: Interview Preparation**
- **Part II: Competitive Programming**
- **Part III: Behavioral Interview (must for any SWE)**
- **Part IV: System Design Interview (> entry level)**
- **Part V: Machine Learning Interview (ML Engineer/Data Scientist)**

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
 - Disjoint Set Union

Interview Topics

- **Real Differentiators (Tech vs Quant)**

- **Strings:** Knuth Morris Pratt (KMP); Rabin Karp / String Hashing; Suffix Array; Suffix Automaton;
- **Data Structures:** Segment Tree; Fenwick Tree;
- **Dynamic Programming:** 1D; 2D; Interval; Tree;
- **Greedy Algorithms** and Invariants: Matroid;
- **Graphs:** Shortest Path; Flow / Matching; Minimum Spanning Tree;

Interview Topics

- **Real Differentiators (Tech vs Quant)**

- **Strings:** Knuth Morris Pratt (KMP); Rabin Karp / String Hashing; Suffix Array; Suffix Automaton;
- **Data Structures:** Segment Tree; Fenwick Tree;
- **Dynamic Programming:** 1D; 2D; Interval; Tree;
- **Greedy Algorithms** and Invariants: Matroid;
- **Graphs:** Shortest Path; Flow / Matching; Minimum Spanning Tree;
 - SP: BFS/DFS; 0-1 BFS; Dijkstra; Bellman-Ford; Floyd-Warshall;
 - Flow: Edmond-Karp + Ford-Fulkerson / Dinic; MCMF;
 - MST: Kruskal; Prim;

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];  
}
```

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!

- **Annual Contests**

- Meta Hacker Cup

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!**

- **Annual Contests**

- Meta Hacker Cup

Interview Preparation Resources

- **Elements of Programming Interview (Tech)**
- **Competitive Programming 4 (Quant)**

Part II: Competitive Programming

Programming Zealots @Discord

- Break into CodeForces rating of **2200+** as fast as you can!
- Join discord, if you have not already!!!

<https://bit.ly/programming-zealot>



Programming Zealots @CodeForces

- Also, join CodeForces group, if you have not already!!!

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 the ICPC North America Championship (NAC)!

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 the ICPC North America Championship (NAC)!
- 2200 - 3500 (O - ZB)
 - **For those who are more serious**
 - To make it to the ICPC World Finals (and potentially winning a medal)!
 - In **Part II** at 4pm today, we will dive deeper into this!

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!

International Collegiate Programming Contest (ICPC)

- If you would like to get involved in helping out as a volunteer or an official intern (unpaid), please reach out to me at christian.lim@icpc.global.
- Or, book a 1:1 time slot using bit.ly/yongwhan-quickchat



Part III: Behavioral

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

- What experiences do you have relevant to this job?

Example Question #2: 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 JetBrains vs ?

Resources

- There are number of preparation books.
- For example:
 - *Behavioral Interview Questions and Answers* by Horatio Bird;
 - For entry level+
 - *Leadership Interview Questions You'll Likely Be Asked* by Vibrant Publishers;
 - For senior level+ (5 years+ of experience)

Part IV: System Design

System Design Interview

- 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

Part V: Machine Learning

Machine Learning Interview

- **Hands-on Experience** using TensorFlow/Keras/PyTorch: comfortable using data to feed into a baseline model.
- **ML Foundations** (e.g., linear regression, support vector machine, etc.)
- **Recent Trends** (reinforcement learning, deep learning architectures, etc.)

Machine Learning Interview

- **Hands-on Experience** using TensorFlow/Keras/PyTorch: comfortable using data to feed into a baseline model.
- **ML Foundations** (e.g., linear regression, support vector machine, etc.)
- **Recent Trends** (reinforcement learning, deep learning architectures, etc.)

- **In-depth knowledge** of a specialization (e.g., computer vision) can be a plus, but not required.

Example Question (Theory)

- What is a difference between unsupervised learning and supervised learning? What are some examples of each?

Example Question (Hands-on)

- What are some practical ways to avoid overfitting? Have you implemented some of those techniques before?

Example Question (Implementation)

- Given a stock market data, predict the future stock price. What are some different approaches here?

(Must!) Resources

- **Textbooks:** *Deep Learning* by Ian Goodfellow, et. al.
- **Courses:** Stanford CS 229 (Machine Learning); ...
- **Tools:** PyTorch; Keras; TensorFlow; Jupyter; ...

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>



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! :)