University of Calgary
Technical Interview Workshop

Christian Yongwhan Lim

4pm MT, November 10, 2023

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:
  - **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;
  - **Visiting Instructor** at Cornell-Tech;

https://www.yongwhan.io

NOW: it's YOUR turn!

- Please fill out this form: https://bit.ly/calgary-tech-workshop

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

Interview Topics

- **Real Differentiators (Tech vs Quant)**
  - Strings
  - Dynamic Programming
  - Greedy Algorithms and Invariants
  - Graphs

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;

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

Part II: Competitive Programming

CodeForces

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

- Join the training sessions through **Programming Zealots** (**https://bit.ly/programming-zealot**).

Success Pathways

- Those who are just starting should focus on the **first half** of problems in Zealot Problem Set. Your main focus should be gaining some experiences with an explicit goal to enjoy the process of solving new problems and potentially making it to the ICPC North America Championship (NAC)!

- Those who are more serious should focus on the **second half** of problems in Zealot Problem Set. Your goal should be making into the World Finals and potentially winning a medal!

Practice Strategies

- 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}.
- **(Eventual) Target**: You should focus on solving it for 30 minutes or less!

Practice Strategies

- 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:

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.

International Collegiate Programming Contest (ICPC)

- If you would like to get involved in helping out as a volunteer or an official (unpaid) intern, please reach out to me at **christian.lim@icpc.global**.

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 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;

Part IV: System Design

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

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 Questions

- [**Theory**] What is a difference between unsupervised learning and supervised learning?

Example Questions

- [**Theory**] What is a difference between unsupervised learning and supervised learning?

- [**Hands-On**] What are some practical ways to avoid overfitting?

Example Questions

- [**Theory**] What is a difference between unsupervised learning and supervised learning?

- [**Hands-On**] What are some practical ways to avoid overfitting?

- [**Implementation**] Given a stock market data, predict the future price.

(Must!) Resources

- **Textbooks**: *Deep Learning* by Ian Goodfellow, et. al.

- **Courses**: Stanford CS 229 (Machine Learning); ...

- **Tools**: PyTorch; Keras; TensorFlow; Jupyter; ...

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