TWO SIGMA

10:30am PT, Friday, November 18, 2022

Technical Interview Workshop @ USC

Yongwhan Lim
Senior Quantitative Software Engineer

Important Legal Information

This document is being distributed for informational and educational purposes only and is not an offer to sell or the solicitation of an offer to buy any securities or other instruments. The information contained herein is not intended to provide, and should not be relied upon for, investment advice. The views expressed herein are not necessarily the views of Two Sigma Investments, LP or any of its affiliates (collectively, "Two Sigma"). Such views reflect the assumptions of the author(s) of the document and are subject to change without notice. The document may employ data derived from third-party sources. No representation is made by Two Sigma as to the accuracy of such information and the use of such information in no way implies an endorsement of the source of such information or its validity.

Disclaimer

The copyrights and/or trademarks in some of the images, logos or other material used herein may be owned by entities other than Two Sigma. If so, such copyrights and/or trademarks are most likely owned by the entity that created the material and are used purely for identification and comment as fair use under international copyright and/or trademark laws. Use of such image, copyright or trademark does not imply any association with such organization (or endorsement of such organization) by Two Sigma, nor vice versa.

Yongwhan Lim

Senior Quantitative Software Engineer at Two Sigma

Education`

Stanford University

MIT

Part-time Jobs

Cornell Tech

MIT EECS

Full-time Job

Google Research

TWO SIGMA

Workshops

Stanford Engineering | Stanford Computer Forum

Carnegie Mellon University

Harvard

University of California

KAIST

NUS National University of Singapore

Coach/Judge

icpc International Collegiate Programming Contest

icpc.foundation

advancing the art and sport of competitive programming

https://www.yongwhan.io/

Overview

- 10:30AM Session: Entry Level

- 12PM Session: Experienced Level

10:30AM Session
Entry Level

Overview

- Part I
  - Interview Types
  - Technical Interview
  - Interview Topics
  - 2 Sample Interview Questions
  - Interview Preparation Resources

- Part II: Questions and Answers (Q&A's)

Part I

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 Overview (Company Dependent)

- Recruiter Call

- 0-1 Online Coding Challenge
  - automated screening with 2-3 questions.

- 2-3 Technical Phone Screenings
  - first technical conversation with human.

- 4-7 Interviews Onsite
  - similar to phone screening but more in-depth.
  - you may get probed on your claimed expertise.

- 0-5 Fit Calls and Negotiation

Technical Interview Overview (Company Dependent)

- Recruiter Call

- **0-1 Online Coding Challenge**
  - automated screening with 2-3 questions.

- **2-3 Technical Phone Screenings**
  - first technical conversation with human.

- **4-7 Interviews Onsite**
  - similar to phone screening but more in-depth.
  - you may get probed on your claimed expertise.

- 0-5 Fit Calls and Negotiation

Interview Topics Overview

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

Interview Topics Overview

- **Data Structures and Algorithms**

- (> entry level) System Design Problems

Interview Topics Overview

- **Fundamentals**
  - Primitive Types
  - Arrays and Linked Lists
  - Binary Trees
  - Heaps
  - Sorting
- **Important**
  - Stacks and Queues
  - Hash Tables
  - Binary Search Trees
  - Searching
  - Recursion
- **Real Differentiators**
  - Strings
  - Dynamic Programming
  - Greedy Algorithms and Invariants
  - Graphs

Interview Topics Overview

- **Fundamentals**
  - Primitive Types
  - Arrays and Linked Lists
  - Binary Trees
  - Heaps
  - Sorting
- **Important**
  - Stacks and Queues
  - Hash Tables
  - Binary Search Trees
  - Searching
  - Recursion
- **Real Differentiators**
  - Strings
  - Dynamic Programming
  - Greedy Algorithms and Invariants
  - Graphs

Sample Interview Question #1

- **Problem Statement** (LeetCode #1201: Medium)
  - An **ugly number** is a positive integer that is divisible by a, b, or c.
  - Given four integers n, a, b, and c, return n^{th} **ugly number**.

Sample Interview Question #1

- **Problem Statement** (LeetCode #1201: Medium)
  - An **ugly number** is a positive integer that is divisible by a, b, or c.
  - Given four integers n, a, b, and c, return n^{th} **ugly number**.

- **Constraints**
  - n, a, b, c ≤ 1,000,000,000

Sample Interview Question #1

- **Problem Statement** (LeetCode #1201: Medium)
  - An **ugly number** is a positive integer that is divisible by a, b, or c.
  - Given four integers n, a, b, and c, return n^{th} **ugly number**.

- **Constraints**
  - n, a, b, c ≤ 1,000,000,000

Ideas?

Sample Interview Question #1

- **Binary Search Solution (Logarithmic):**

Now, do you see it?

Sample Interview Question #1

- **Binary Search Solution (Logarithmic):**

```
#include<bits/stdc++.h>
using namespace std;

int nthUglyNumber(int n, int a, int b, int c) {
        int low = 1, high = INT_MAX;
        while(low < high) {
                int mid = low + ((high - low) >> 1);
                if(eval(mid, a, b, c) >= n) {
                        high = mid;
                } else {
                        low = mid + 1;
                }
        }
        return low;
}
```

```
typedef long long ll;

ll lcm(ll a, ll b) {
        return a/__gcd(a,b)*b;
}

ll eval(ll x, ll a, ll b, ll c) {
        return x/a + x/b + x/c - x/lcm(a,b) - x/lcm(a,c) - x/lcm(b,c) + x/lcm(a,lcm(b,c));
}
```

Sample Interview Question #2

- **Problem Statement** (LeetCode #1312: Hard)
  - Given a string s. In one step you can insert any character at any index of the string.
  - Return *the minimum number of steps* to make s palindrome.
  - A **Palindrome String** is one that reads the same backward as well as forward.

Sample Interview Question #2

- **Problem Statement** (LeetCode #1312: Hard)
  - Given a string s. In one step you can insert any character at any index of the string.
  - Return *the minimum number of steps* to make s palindrome.
  - A **Palindrome String** is one that reads the same backward as well as forward.

- **Constraints**
  - $1 \leq |s| \leq 500$
  - s consists of lowercase English letters.

Sample Interview Question #2

- **Problem Statement** (LeetCode #1312: Hard)
  - Given a string s. In one step you can insert any character at any index of the string.
  - Return *the minimum number of steps* to make s palindrome.
  - A **Palindrome String** is one that reads the same backward as well as forward.

- **Constraints**
  - $1 \leq |s| \leq 500$
  - s consists of lowercase English letters.

Ideas?

Sample Interview Question #2

- **Dynamic Programming Solution (Quadratic):**

Now, do you see it?

Sample Interview Question #2

- **Dynamic Programming Solution (Quadratic):**

```
#include<bits/stdc++.h>
using namespace std;

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

- **Popular Websites**
  - LeetCode
  - CodeForces
  - AtCoder
  - TopCoder
  - CodeChef

Interview Preparation Resources

- **Popular Websites**
  - **LeetCode**
  - CodeForces
  - AtCoder
  - TopCoder
  - CodeChef

Interview Preparation Resources

- **Popular Websites**
  - **LeetCode**
  - CodeForces
  - AtCoder
  - TopCoder
  - CodeChef

- Try to solve all problems from biweekly/weekly LeetCode contest **fast**.
  - Here, fast means under 1 hour for all four questions!

- Aim to be on **division I** at CodeForces
  - This will trivialize most of the technical interview.

Interview Preparation Resources

- Standard
  - *Elements of Programming Interview* by Adnan Aziz, et. al.
  - *Cracking the Coding Interview* by Gayle Laakmann McDowell

- Overkill
  - *Competitive Programming 4* by Steven Halim, et. al.
  - *Guide to Competitive Programming* by Antti Laaksonen

Interview Preparation Resources

- Standard
  - ***Elements of Programming Interview*** by Adnan Aziz, et. al.
  - ***Cracking the Coding Interview*** by Gayle Laakmann McDowell

- Overkill
  - *Competitive Programming 4* by Steven Halim, et. al.
  - *Guide to Competitive Programming* by Antti Laaksonen

Part II
Q & A's

Q & A's

- How do you overcome nervousness?

Q & A's

- Could you provide a live solving of a technical question?

Q & A's

- Is interview process as an intern different from full-time technical interview?

Q & A's

- How do you get past the automatic filter?

Q & A's

- What are topics to prepare, the best way to prepare, and expectations on programming language?

Q & A's

- Will interviewer evaluate applicants' technical knowledge other than coding skills?

Q & A's

- Are there any specific machine learning and artificial intelligence technical questions that frequently show up in interviews (and that we should prepare for)?

Q & A's

- If I do not have too much background on a position I am applying for, how do I leave a good impression to interviewer still?

12PM Session
Experienced Level

Overview

- Part I

  Behavioral interview (must for any SWE)
  System design interview (> entry level)
  Machine learning interview (ML engineer)

- Part II: Questions and Answers (Q&A's)

Part I

Part I-A
Behavioral

Behavioral Interview (for everyone)

- Becoming an industry standard to have **at least one** session in typical Software Engineer (SWE) 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, databases, algorithmics, development tools, etc.)

Resources

- There are number of preparation books.

- For example:

  Behavioral Interview Questions and Answers by Horatio Bird.
  Leadership Interview Questions You'll Most Likely Be Asked by Vibrant Publishers

Part I-B
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.

Example Question #1

- Design YouTube

Example Question #2

- Design Instagram

Resources

- Must reads are:

  The System Design Interview, 2nd edition by Lewis C. Lin, et. al.
  System Design Interview by Alex Xu

Resources

- Must reads are:

  The System Design Interview, 2nd edition by Lewis C. Lin, et. al.
  System Design Interview by Alex Xu

- If you have time to dig deeper, consider:

  Understanding Distributed Systems by Roberto Vitillo
  Designing Data-Intensive Applications by Martin Kleppmann

Part I-C
Machine Learning

Machine Learning Interview (for ML SWE)

- **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 (for ML SWE)

- **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 can be a plus, but not required (e.g., computer vision, natural language processing, etc.).

Example Question (Theory)

- What is a difference between unsupervised learning and supervised learning?

Example Question (Hands-on)

- How do you avoid overfitting?

Example Question (Implementation)

- Given stock market data, predict the future stock price.

(Must!) Resources

- **Textbooks**

  Deep Learning by Ian Goodfellow, et. al.

- **Courses**

  CS 229 (Stanford): Machine Learning

- **Tools**

  PyTorch
  Keras
  TensorFlow
  Jupyter

Part II
Q & A's

Q & A's

- What are some tips to be successful in the interview process at big techs?

Q & A's

- What are some good questions to ask after the interview?

Q & A's

- How should I communicate with interviewer during interview?

Q & A's

- What are the good and bad examples of a technical interview?

Q & A's

- Is LeetCode enough or preparing the technical interview?

Q & A's

- What do you do if you do not know how to solve or answer a problem?

Q & A's

- What are some typical technologies you must know to succeed in a technical interview?

Q & A's

- (COVID) How is a virtual technical interview different from the in-person interview?

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!

- **1:1 Meeting Opportunity**: https://calendly.com/yongwhan/one-on-one/